CSDN 技术改变世界 创新驱动中国

Programmer 程序员

ISSN 1672-3252

2011年 1月刊

邮发代号：2-665 定价：15元




谷歌中国工程研究总经理 **杨文洛**
互联网发展中的三大挑战

Parasoft公司总经理 **陈兆仁**
自动化测试+完善流程=抢得先机




Frost & Sullivan中国区总裁 **王煜全**
用户的需求从哪里来

NHN中国技术开发中心总监 **崔海**
信任的力量



成功产品三要素

变革中的技术 • 用户为中心的物联网系统设计
务实的用户体验设计 • 交互设计师的核心竞争力
多彩的产品之年 • 我们到底是不是产品经理
用人类行为学理论指导Web设计

P81 丁香园网站CTO 冯大辉 十年莽撞T路
P99 Java技术权威Joshua Bloch专访 Java的命运
P42 2010软件开发2.0大会特别专题 中国开发者将迎来黄金十年
架构师接龙 盛大许式伟 VS. 金山张宴 P90
关于MySQL多机房部署那点事儿 P106
程序员是这个时代的思想家 P135
分布式流式计算平台—S4 P103

程序员 Programmer 2011年 01月刊 技术改变世界 创新驱动中国

特别感谢

(排名不分先后，以姓氏首字母为序)

Contributors



陈杰

感谢猫扑网技术总监陈杰接受本刊采访，详见本期对话 CTO 栏目。



陈兆仁

感谢 Parasoft 公司亚太区总经理陈兆仁分享如何提高软件质量的观点，详见“高端视点”栏目。



崔海

感谢 NHN 中国技术开发中心总监崔海接受本刊采访，详见本期报道。



高博

感谢盛大创新院高级研究员高博接受本刊采访，分享产品方面的见解，详见“封面报道”栏目。



胡宁

感谢 MediaV CTO 胡宁接受本刊采访，详见本期对话 CTO 栏目。



黄晓东

感谢移动立方创始人黄晓东接受本刊采访，详见本期“一分钟先生”专栏。



解研科

感谢百度解研科接受本刊采访，分享用户体验方面的见解，详见“封面报道”栏目。



柯博文

感谢美国硅谷 LoopTek 公司 CEO 柯博文评析互联网电视的现状，详见“产品报道”栏目。



刘大卫

感谢街旁网 CEO 刘大卫为本刊撰稿，详见本期“一分钟先生”专栏。



吕建伟

感谢《走出软件作坊》图书作者吕建伟为本刊撰稿，详见本期“一分钟先生”专栏。



马如悦

感谢百度基础架构部高级工程师马如悦为本刊撰文，详见“技术”栏目。



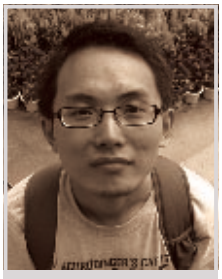
毛新生

感谢 IBM 中国新技术研发中心总经理、IBM 中国开发中心 CTO 毛新生接受本刊采访，分享技术方面的见解，详见“封面报道”栏目。



潘爱民

感谢知名图书作者、盛大创新院潘爱民接受本刊采访，详见本期“一分钟先生”专栏。



苏杰

感谢阿里巴巴产品经理、《人人都是产品经理》作者苏杰对本刊的支持，详见“封面报道”栏目。



覃旋

感谢中游在线技术顾问覃旋提供 RIA 工具的对比，详见“产品报道”栏目。



王坚

感谢糗事百科创始人王坚接受本刊采访，分享产品方面的见解，详见“封面报道”栏目。



王亚洲

感谢 SAP 中国研究院王亚洲为本刊撰文，分享用户体验设计方面的收获，详见“封面报道”栏目。



吴琼

感谢 Autodesk 中国研究院吴琼接受本刊采访，分享用户体验方面的见解，详见“封面报道”栏目。



西乔

感谢《神秘的程序员们》漫画作者、Magicome 公司创始人西乔接受本刊采访，详见“封面报道”栏目。



杨卫华

感谢新浪产品事业部技术经理杨卫华接受本刊采访，分享技术方面的见解，详见“封面报道”栏目。



杨文洛

感谢谷歌中国工程研究总经理杨文洛分享互联网发展的观点，详见“高端视点”栏目。



杨武

感谢思科系统中国研发有限公司软件工程师杨武接受本刊采访，分享产品方面的见解，详见“封面报道”栏目。



叶忻

感谢架势无线 CEO 叶忻接受本刊采访，详见本期“一分钟先生”专栏。



张宴

感谢金山游戏运营技术中心技术经理、专家组成员张宴为本刊撰文，详见“架构师接龙”栏目。

解读大趋势

辞旧迎新之际，理所应当做点回顾和展望的工作。更何况，我们正处在这样一个关键时刻。过去一年中，业内的思想者们都不约而同地在提醒我们：大变革已经开始。

2010年2月微软负责软件开发者部门的全球高级副总裁S.Somasegar发表题为《软件开发关键趋势》的文章，指出今天的开发者比以往任何时候都面临更多的技术选择，他认为云计算、以Web为平台、并行计算、设备和用户界面多样化、敏捷开发过程、异地协同开发是其中最为重要的六大趋势。

10月Gartner发布报告，将云计算、移动应用和媒体平板、社会交流和协作、视频、下一代分析技术、社会化分析、情景感知计算、存储级内存、泛在计算（相当于国内所说的物联网）、模块化（fabric-based）的IT基础设施选为2011年十大战略性技术。

同月，另一家知名的顾问公司Forrester也发布了2011–2013年技术趋势报告，列举了15个企业架构师需要关注的技术趋势。其中，将SaaS和云计算平台成为标准、协作平台以人为本、客户社区平台与业务应用集成、应用与业务流程移动化、网真系统广泛应用五方面列为关键技术趋势。其他十大趋势是：下一代商务智能与实时、泛在和灵活访问结合、针对文本和社会化网络的分析、IaaS应用增加、主数据管理成熟代表的以流程为中心的数据与智能，业务规则处理和机遇策略的SOA进入主流、支持Web 2.0的业务流程管理、事件驱动模式获得更多关注为代表的敏捷应用，系统管理支持持续虚拟化、IT通过规划和分析工具来管理未来发展、客户端虚拟化无所不在代表的智能管理。

透过这些林林总总的分析和预测，我们可以清楚地看到一些共识：**云计算**（Web平台、分布式计算、海量数据存储与处理、虚拟化、灵活的IT基础设施和流程管理）、**社会化**（协作、社区、敏捷）、**移动**（多终端、全新用户体验、实时、位置）、**物联网**（泛在计算、增强现实）、**分析**（数据挖掘、商务智能）。这些关键词共同组成了这几年的技术宏观图景。

过去一年给我留下最深印象的，还是前微软首席架构师Ray Ozzie在10月辞职后发表的长文“Dawn of a New Day”（新时代的曙光）中发出的警世之语：**持续在线设备和持续服务将带来史无前例的巨大变革，不仅是互联网行业，所有行业现在都应该退一步，质疑自己最本质的东西，重新思考应对策略。**

让我们听一听Ozzie这位我们时代数一数二的技术先知是怎么说的吧。

> 各种业务流程乃至整个公司都已经被互联网零阻尼的本质所改变，制造者与消费者之间的墙现在消失了。实体业务生态系统已经坍塌，因为传统的聚合与分布机制不再有效了。
>
> 今天，全世界所有行业的公司都在向后退一步，重新思考基本的东西，质疑他们最基本的结构性根基。这对于公司长期的成长和生存是必不可少的。我们自己的行业也不例外，过去关于基础设施和应用最基本的假设，都必须进行重新判定。
>
> 过去五年是激动人心的，但是未来五年将迎来另一个拐点——在应用和基础设施上巨大和不可避免的转变所激发的大变革现在才刚刚开始，它将给我们行业带来史无前例的机遇。

我们的未来——社会、国家、政府、行业、工作和生活会变成怎样？现在就退后一步，清空成见，回归本质，重新思考吧。欢迎与我们分享你的心得。





博客：blog.csdn.net/liujiangce

Contents

50 封面报道 Cover Story



今天，产品已经成为市场竞争的核心，产品的成败关乎研发人员、团队乃至整个企业的兴衰。要想做出一款成功的产品，必须具备三个基本要素：技术（Technology）、用户体验（User Experience）、产品运营管理（Product），我们将之简称为TUP。技术显然是成功产品的必备条件，用户体验对于产品成功的重要性更是不言而喻，而如何对产品进行营销和管理也是一门艺术。本期“封面报道”，为你诠释TUP的真谛。

52 变革中的技术
——2010–2011回顾与展望
56 用户为中心的物联网系统设计
60 务实的用户体验设计
——用户体验设计的2010年
63 用人类行为学理论指导Web设计
67 交互设计师的核心竞争力
71 多彩的产品之年
——产品经理一席谈
74 我们到底是不是产品经理
78 2010年度CSDN十大资讯（编辑推荐）
79 2010年度CSDN十大博客文章
80 2010年度CSDN十大热点资讯

固定专栏 Columns
7 名人堂 Hall of Fame
12 外刊速递 Abroad Media
14 微博 Micro-Blogging
16 程序天下事 Technical News
136 漫画与幽默 Humor

企业专栏 Business
96 模型即软件，建模即开发
——顶点软件LiveBOS灵动业务架构平台

高端视点 Leaders' Viewpoint
8 自动化测试+完善流程=抢得先机
10 互联网发展中的三大挑战

报道 Report
30 技术社区的新发展
31 TechED China 2010，“云”的大会
32 新产品新技术新趋势
——记甲骨文全球大会
34 信任的力量
——走进NHN 中国技术开发中心
37 Office系统对信息工作人员价值最大
——微软MVP周庆麟专访

对话CTO CTO Columns
38 互联网不适合螺丝钉员工
39 技术创新是从业准则

2010软件开发2.0大会特别专题
12月9日~10日，由CSDN主办，盛大网络、《程序员》杂志、JavaEye社区协办的软件开发2.0大会成功举办，来自阿里巴巴、盛大、微软、Facebook等公司的技术领袖们汇聚一堂，共同剖析当前IT技术的发展大势。
42 中国开发者将迎来黄金十年
——2010软件开发2.0大会报道
45 架构重构趋势谈
48 软件企业常见问题和系统性解决方法

程序人生 Coding Life
81 十年莽撞IT路
丁香园网站CTO冯大辉讲述精彩程序人生。

软件工程 Software Engineering
86 这个圈圈不简单
用例背后隐含的是市场的思维，对仍习惯于学生思维的开发人员来说，用例是一种折磨。

架构 Architecture
90 许式伟VS.张宴
怎样保证网站的高可用性？如何做好任务切分？如何保证产品质量？面试会问哪些问题？好戏不断，尽在本期架构师接龙栏目。

Contents

2011年1月刊　总第219期

一分钟先生 Mr. One Minute

94 我的2010年

技术 Technology

99 Java的命运

本文是Common Lisp专家Peter Seibel对Google公司首席Java架构师Joshua Bloch的访谈，讲述他所遇到的最糟糕的Bug及其对Java命运的看法。

103 分布式流式计算平台——S4

本文是作者在充分阅读和理解Yahoo！最新发布的技术论文《S4:Distributed Stream Computing Platform》基础上，所做出的知识分享。

106 关于MySQL多机房部署那点事儿

MySQL多机房部署中存在哪些问题？应该怎样合理解决？本文作者给出了精彩解答。

109 基于反射的软件自动化测试框架设计（下）

文章以Windows SDK工具箱中Service Configuration Editor的实际测试为例，演示了反射在测试中的应用。

113 云计算的数据安全

本文从数据隐私和数据隔离两个技术角度，对云计算中的数据安全进行了阐释和剖析。

移动专栏 Mobile

116 移动游戏架构的艺术

游戏是一门综合的艺术，作者以铸剑为喻，形象地展示了移动游戏架构的艺术之美。

调试之剑 Debugging Sword

119 趣谈CLR4的调试模型重构（上）

工具点评 Tools Reviews

122 iPad上的设计工具

产品报道 Products Report

124 三网融合下的国外互动式电视新契机

126 主流RIA技术分析与对比

产品推荐 Products Recommendation

128 新产品新工具

130 Geek产品

图书 Books

132 新书上架

评论 Comments

134 用户的需求从哪里来

135 程序员是这个时代的思想家

主管：中国社会科学院
主办：中国社会科学院文献信息中心
出版：《程序员》杂志社
网址：http://www.programmer.com.cn
国际刊号：ISSN 1672-3252
国内刊号：CN11-5038/G2
邮发代号：2-665
广告经营许可证号：京东工商广字0188号

总编：黄长著　Editor-in-chief：Huang Changzhu
社长/常务副总编：张悦校　President：Zhang Yuexiao
副社长：蒋 涛　Vice President：Jiang Tao
编委会：黄长著　张悦校　陈洋彬　蒋涛　曾登高　刘江
Editorial Member：Huang Changzhu　Zhang Yuexiao　Chen Yangbin　Jiang Tao
Zeng Denggao　Liu Jiang

执行主编：孟迎霞　Executive Editor-in-chief：Meng Yingxia
编辑部主任：常政　Director：Chang Zheng
责任编辑：郑柯　董世晓　高松
Editors：Zheng Ke　Dong Shixiao　Gao Song
特邀编辑：方梁　高昂　赵健平　吕娜　卢鹄翔
Contributing Editors：Fang Liang　Gao Aang　Zhao Jianping
Lv Na　Lu Dongxiang
美术设计：纪明超　Art Designer：Ji Mingchao
美术编辑：林象海　Art Editor：Lin Xianghai
流程编辑：吴志民　Coordinator：Wu Zhimin
Tel：010-64351458
Email：editor@csdn.net

发行部　Distribution Dept.010-64351431
Email：sales@csdn.net

广告总代理：北京创新乐知广告有限公司
Sole Advertising Agency：Bejing CSDN Co.,Ltd
Tel：010-64376055
Email：ad@csdn.net
Marketing Dept：010-51661202（ext 149）
Email：market@csdn.net

读者服务部
Readers service Dept.
网上订购：http://dingyue.programmer.com.cn/
读者信箱：reader@csdn.net
地址：北京市朝阳区广顺北大街33号院1号楼福码大厦B座12层
Address: B-12th Floor Fairmont Tower NO.33 Guangshun North street,
Chaoyang District,Beijing
邮政编码：100102
电话：010-64351436
传真：010-64348545

法律顾问：北京中润律师事务所 王杰
Law Consultant：Beijng Hengsheng Lawyer Firm
印刷：北京盛通印刷股份有限公司
Print: Beijing Shengtong Printing Co., Ltd.
出版日期：每月1日
Publication Date：the first day per month
零售价：RMB 15.00元　新台币 390元　HK $ 35.00（港、澳）
US $ 9.00（海外）
Retail Price：RMB 15, NT$390, HK $ 35.00,US $ 9.00

本刊文章版权所有 未经许可不得转载
发现装订错误或缺页，请将杂志寄回本刊读者服务部，即可得到调换。

计算机科学的奠基人

——Alan Perlis

■ 文 / 苏椰



当我们提起高级语言的先驱，首先想到的总是Dennis Ritchie、Bjarne Stroustrup这些主流语言的设计者，似乎很少会提到Alan Perlis这个名字。但Alan Perlis主持设计的ALGOL，趋势C/C++等语言的鼻祖。

Alan Perlis 1922年出生于美国匹兹堡犹太家庭。21岁时，他获得了卡内基理工学院的化学学士学位，这个学院后来发展为现在美国计算机专业排名第一的卡内基梅隆大学。时逢二战，他弃笔从戎。服役期间，他突然对数学产生了强烈兴趣，并在1950年，从麻省理工学院获得了数学博士学位。两年后他来到普渡大学，出任普渡大学计算中心首任主任，并将IBM CPC计算机引入了普渡，还为其设计了一个叫作IT（Internal Translator）的编程语言。四年之后，他又一跃当上了卡内基理工学院第一任计算机科学系主任，在卡耐基引入了IBM 650，并将IT语言移植到650上。这些使他在程序语言的设计方面，有了许多经验和体会。当ACM成立程序设计语言委员会时，Perlis坐上了主席的位置。

1958年5月27日，瑞士苏黎世召开了一场8个人的讨论会。这个会议并不隆重。ACM的4名代表和德国应用数学和力学学会（GAMM）的4名代表，索性把地点和双方名称连起来，叫苏黎世ACM-GAMM会议。这场会议的组织者，就是Alan Perlis。他们将要在此讨论并规划一种新型的编程语言，叫作国际代数语言（IAL）。

世界上第一个高级语言Fortran存在一些严重的缺陷。比如说它专门为IBM 704设计的，要依赖特定的机器型号工作，很难向其他机器移植等等。IAL的专家们力求设计一种更好的高级语言。在讨论过程中，Perlis认为“IAL”这个词很绕口，于是将它改名叫作ALGOL。会议结束后，他们成立了一个工作组，根据讨论的结果，开发ALGOL的编译器。1958年年底，第一套编译器诞生了，按照年份命名为ALGOL 58。1960年，Alan Perlis总结了一些经验之后，再次召集参与ALGOL工作的计算机科学家们在巴黎进行了另一场研讨。这场会议的结果，就是后来的ALGOL 60。

1960年夏天，Dijkstra开发了第一个ALGOL 60的编译器，于是ALGOL系列语言正式登上了计算机科学的舞台。随后，Alan Perlis将它引入了大学的课堂。这是Alan Perlis的另一个重要贡献，因为在那个时代，计算机科学混沌初开，程序设计都只是数值分析的一部分。经过Alan Perlis的努力，人们终于看到计算机科学应该是什么样子。Dijkstra后来说，这是一个伟大的标志，可以说直到这个时候，计算机科学才真正地诞生了。而在接下来的30年里，ALGOL一直是教学和学术界用来描述算法的不二之选。它体现出的许多概念，都被后来的编程语言沿用。包括C、C++和Pascal在内许多主流语言，都因为继承了ALGOL的许多概念，而被称为“类ALGOL语言”。

Alan Perlis因领导了ALGOL的设计工作，并在早期计算机教育中做出了重要贡献，在1962年当选为美国计算机学会的主席，并在四年后，因为他对高级编程技术及其编译器构造的影响成为历史上首位图灵奖得主。五年后，他跳槽到耶鲁大学，连任多年计算机科学系主任，1977年当选美国工程院院士。1982年，退休后的Alan Perlis返老还童，写下了名文“Epigrams on Programming”，发表在ACM的SIGPLAN期刊上。其中包含130条编程箴言，可能是被引用最广泛的计算机文章之一。在发表在ACM的SIGPLAN期刊上，他写下了一系列关于编程的幽默：“如果你给别人讲解程序时，看到对方点头了，那你就拍他一下，他肯定睡觉了。”

Alan Perlis为名著《计算机程序的构造和解释》所写的序中写道，如果说艺术解释了我们的梦想，那么计算机就是以程序的名义执行着它们。计算机科学中特别重要的一点是保持趣味性，不要局限于眼前，应该不断寻找新方向，扩展计算机的能力。这些话今天听来，依然发人深省。

1990年2月7日，Alan Perlis因心脏病在纽哈芬去世，享年68岁。

自动化测试+完善流程=抢得先机



陈兆仁
Parasoft公司亚太区总经理

Parasoft公司1988年成立于美国加州蒙罗维亚，二十多年来它一直专注于两件事：一是如何用自动化的方式提高软件企业的生产效率；二是让需要很多人力、时间的测试工作以自动化的方式达成。在每一个领域，无论是嵌入式系统、Web Service、Cloud、SOA，或者是主机平台的开发，Parasoft都有相应的产品去协助使用者，让开发更有效率，自动化测试更完整。

自动化测试领域的努力

Parasoft在自动化测试领域的努力包括以下几个方面。

测试层级的角度，从最早的静态分析、资料流分析，到单元测试、集成测试、安全测试、压力测试，Parasoft都能够协助不同的开发人员自动化这些部分。

编程语言层级的角度，对于J2EE、ESB、C和嵌入式C、.NET或C#等不同的编程语言，Parasoft都有相应的工具去协助开发者完成测试需求。

系统层级的角度，比如针对一个银行系统的测试，包含了J2EE，ESB等不同的技术在里面，还有新的SOA系统，大量使用Web Service，而且常常是相对比较庞大，可能被不同的部门、上下游厂商甚至用户使用，常常是牵一发而动全身。因此，测试方面复杂度就增加了非常多，安全性的需求也变得非常多，Parasoft也有相应的自动化工具去协助，能够更快速地达到一些指标。

完善流程的努力

Parasoft在完善流程方面也不曾松懈。我们知道在软件开发生命周期里面，宏观来讲遇到的问题不外乎有四项：否能及时交付；否能在预算内交付；否能实现预期功能；否能正常运作。Parasoft要解决这四个问题，最重要的是有更好的流程管理。因为在软件生命周期里，所有的问题都不是一天，而是经年累月产生的。

好的流程包括如下几点。

首先需要及时性的可视性，能够了解你的项目一分一秒的状况。这样才有对项目的控制，有控制才能有管理。

其次是质量跟踪，了解现在还有多少Bug需要解决，有多少功能没有完成，是否要多加人力在上面，能否及时交付。

没有好的流程系统就没有好的项目。而Parasoft提出的解决方案是帮助用户把流程中项目最及时的状态呈现出来。

在新趋势前面抢得先机

IT领域每天变化非常大，新的概念新的技术都在推陈出新。

举例来说，云计算里面包含的技术非常多，如何在最短时间内实现项目，Parasoft提供了一系列的解决方案。比如说，协助客户运用“云”去虚拟化测试环境，这是Parasoft工具里面一个主要功能。提供虚拟化一些外部需求的环境，比如ERP、CRM或者是另外一个完全不同的平台。当用户的系统越来越复杂，要想搭建和生产环境完完全全一样的测试环境，常常是耗时、耗资巨大的一件事情。因此很多国外的公司都走向云，利用云来搭建测试环境。Parasoft的SOA Test虚拟化的功能就非常容易使用，让客户在很短的时间内搭建与生产环境几乎一样的测试环境，让使用者很快就可以测试自己的产品和系统。

以前业界都认为软件质量是“最好达到”，而并非“必须达到”的要求，只要功能实现就好。可是现在业界有越来越多的标准，比如医疗领域，FDA有一连串的和软件质量相关的硬标准，在汽车电子行业也有类似硬标准。如果我们要在这些领域开发软件产品，就要满足这些标准，同时需要非常多的测试步骤。而Parasoft的许多工具就是为了协助客户做自动化测试，让客户能够一步到位达到业界的一些硬标准。

自动化测试在移动领域的应用也非常广泛，以Google Phone为例，其中有很多静态测试的标准，而应用Parasoft的Jtest工具，就能很好地满足Google Phone静态测试的要求。此外，在移动平台上进行支付，因为与之相连的组件非常多，如果只是负责开发其中的一部分，开发者不可能拿到权限去测试其他有互动的组件的信息。这个时候，虚拟化的测试环境就非常重要。

如今软件开发的新趋势是产品更新非常快，如果因为没有好的开发流程和测试平台，让你的系统一直不稳定，就永远会落在别人后面。反之就占得先机。

互联网发展中的三大挑战



杨文洛
谷歌中国工程研究总经理

我使用Google产品超过10年，这段时间互联网的数据（文字、多媒体等）一直在增加，每天的搜索也在增加。在未来10年，将更加依赖互联网、搜索引擎来帮助生活、工作、学习，但是发展中同样存在着一些挑战。

搜索领域的挑战

第一个是怎样把更多的内容带入互联网。比如前期扫描的书、地图数字化或者视频，这需要不断地努力，同时也是我们的使命。

第二个是让不同的人通过不同的方式接触到信息，能够跨语言、跨文本甚至跨平台搜索。Google的使命是“整合全球信息，使人人皆可访问并从中受益”，这在不同的阶段有不同的挑战。

第三个是搜索怎样变得更智能、更了解用户。10年前，搜索全部是文字的网页链，后来是图列整合搜索，要很聪明地把各种各样的信息（图片、视频等）按重要性排序，慢慢变成用户只需要一个搜索框。今后，怎样更智能、更了解用户的需要，怎样更快地把用户要的信息（有些情况用户不需要提出来，甚至还没有讲完）告诉他。Google会尽量想办法，将“我的朋友喜欢看什么书，喜欢去什么餐厅”这类搜索整合到服务。我们的出发点是通过一个搜索框就能够回答用户所有的问题。

全球化创新模式的挑战

创新不会只停留在一种语言或一个国家，其成果很多是全世界可以共享的。例如把翻译整合到搜索，当时我们看到了中国用户有这样的需要。翻译虽然是一种技术，但它在和用户的交互中扮演了重要的角色。从信息传到数据中心做一次翻译，再把数据传回来，整个周期差不多一秒钟。这是中国的研发团队在减少几十微秒、几百毫秒的努力下争取来的。我们在这里投入大量的精力，发现翻译这个产品对中国的用户很有用，并且在开发过程中发现很多国家都有这个需要，这促使我们把它推广到全部非英文区域和国家。这是从本地看到需求从而推到全世界的很好的例子。这背后是中国团队在努力。当然我们很少会只做中文，而希望能够把中国的或者中文相关的产品做到最好。可以想见，这背后不只是中国的研发团队，而是结合了全世界团队的力量。

开放平台的挑战

12年来，Google的开放策略是完全一致的，它的开放程度比10年前、5年前的想象还要更彻底。从先发布再开放（例如Android）到还在开发中的版本就已经开放出来了（如Chrome浏览器）；再到还没有发布，就已经开放了，Chrome OS就是一个明显的例子。

开放平台使得工程师做产品时需要考虑两件事情。一是大部分的软件应用开放，包括开放代码或是API；二是个人开发者，例如GAE平台上，开发者可以把应用放在Google上，使用Google的服务器和存储。

这样的平台对工程师是非常好的模式，很容易开发一个新的应用。

Android现在的版本非常稳定。如果没有它，一个厂家需要嵌入式操作系统，就要找专门的人去开发，困难很大。目前，在Android平台上的开发人员越来越多，因为这个平台，全世界都在创新。PC在20世纪80年代到后来发展这么快，因为它除了芯片和操作系统以外，所有的东西都是开放的，把整个产业链带动起来了。现在Android也扮演了这个角色，它的潜能体现在不只是可以用在手机上。

现在Google提供的任何搜索都是开放平台上的资讯。一个具体的数据开放的例子就是Google Trends。（Google非常重视用户的隐私权问题，提供的信息里面不能涉及个人，这些信息可以应用的范围将不牵涉隐私部分）。很多企业在做生意，应该去Google Trends看一下自己的产品在哪个地方可能卖得比较好，做广告的人便知道自己的广告应该投放到哪个国家。

相信很多人也已经应用到很多此类资讯，我个人也很有兴趣关注这一部分，甚至有时候我觉得数据已经在那里，但大家不太会用，好可惜。所以我们要努力克服这个问题，让用户知道，其实有一个很好的开放平台，你已经可以使用了！

IEEE Computer
2010.12

计算金融学



本期的《IEEE Computer》的封面主题是计算金融学。计算金融学是一个跨学科的领域，依靠金融数学、数值方法和计算机仿真来进行交易、套头交易和投资决策。计算金融学也可以用来帮助进行投资组合风险管理。本期中的几篇专题文章提供了对这个领域的一个概览，列举了数据流管理系统的优点，描述了金融机构在应用中使用并行计算可以得到的益处。

《计算金融学》一文对计算金融学进行了介绍。银行和投资基金的竞争力越来越多地基于所使用的定量系统的质量，包括编程技术、分析方法以及诸如金融预测、期权定价和风险管理等应用。这些构成了计算金融学领域的基本要素。

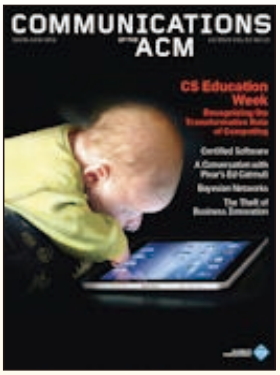
金融应用依赖于时间敏感数据的连续流，任何数据管理系统必须能够瞬间处理复杂的查询。《应用于计算金融学的数据流管理系统》一文指出，尽管许多机构已经转向采用定制的解决方案，数据流管理系统可以提供同样低延迟的处理以及处理一系列应用的灵活性。

通过为应用装备并行计算，金融机构可以得到两个数量级的优化。《华尔街的高性能计算》指出了并行计算在金融领域的应用前景。如果程序员可以应对软件的挑战，在未来几年里，并行化可以显著地改造计算金融学。

《2010-2020年展望》回顾了IEEE计算机协会在2010年一年内的成就，包括会员人数的增加、颁奖典礼的成功举办和社交媒体参与的增长，还有在程序员认证、软件质量、系统和网络安全、在线社区等方面的进展；列举了协会的发展在未来几年内将会面临的挑战，并且展望了下个10年将要走向的目标。

CACM
2010.12

技术前沿



虽然Web 2.0技术确实对企业的基础设施和架构产生了影响，很少有人对Web 2.0技术在企业效率和管理中的贡献进行调查，更没有人以系统的方式研究过它们的使用方式和产生的影响。《Web 2.0技术的商业影响》一文对维基、博客、播客、民众分类、聚合、社交媒体、虚拟世界、众包和RSS过滤器等Web 2.0技术对企业效率和管理的影响进行了调查研究。研究结果显示，维基、博客和RSS过滤器具有的影响最大，虚拟世界具有的影响几乎可以忽略不计。Web 2.0的安全性堪忧，但是沟通和协作普遍得到了加强。由眼睛控制的人机界面已经不是什么新奇的概念了，相关的研究至少可以回溯到20世纪70年代。但是这类界面大规模的应用既非必要，也不为系统设计者们所追捧。《以眼睛为主导》一文中指出，手机的眼球追踪控制可能会打开环境感知用户界面（Context-Aware User Interface）的新纪元。

主题建模是计算机科学、数学和认知科学的综合，可以快速进化以帮助用户理解和导航大量非结构化数据。《主题建模和非结构数据》一文指出使用主题建模，科学家可以探索和理解大量的无标记数据。在伊朗，Stuxnet蠕虫已经感染了3万台设备，包括Bushehr核电站中数量不明的几台设备。这是网络战争还是工业间谍活动？《战争的新面孔》一文对此进行了分析，指出随着复杂的Stuxnet蠕虫的出现，网络战争的界限可以无限地扩展。拼写和语法在编码中是否至关重要？《文学编码》一文作者指出，虽然“程序员不是语言大师”，但是程序员确实需要良好的拼写和编语法来让自己的编码更加容易理解。

MSDN
2010.12

移动应用程序



本期的《MSDN》着眼于Windows Phone 7上的移动应用。在《Windows Phone 7上的数独》一文中，作者用一个基于Silverlight的游戏范例介绍了Windows Phone 7开发中的几个关键概念，如Model-View-View Model（MVVM）设计模式、序列化（Serialization）、用户存储（User Storage）和多方向（Multiple Orientations）。

数据驱动的Windows Phone 7应用程序依赖于良好的UI代码和快速的数据获取。《为Windows Azure和Windows Phone 7建立数据驱动的应用程序》一文对这两个方面进行了深入探讨，并且介绍了如何将Windows Azure与Windows Phone 7开发相结合。

《成功Windows Phone 7应用程序的秘密》一文指出，要想开发出受欢迎的Windows Phone 7应用程序，应把可用性放在第一位。

对很多Silverlight程序员来说，Windows Phone 7最让人振奋的消息就是它将Silverlight作为两种编程接口之一。Silverlight支持灵活的、可触摸的翻阅控制，既可用于Web，也可用于Windows Phone 7。UI前沿专栏文章《Silverlight，Windows Phone 7和多点触摸》一文展示了如何建立一个同时支持这两个平台的共享代码项目，让用户可以通过鼠标和多点触摸进行翻阅。

面向方面是主流的编程范例，尤其胜任将系统分为多个部分，并在各个部分中描述过程的情况。但涉及处理横切关注点（Cross-Cutting Concern）时，面向方面范例不是特别有效。在前沿视角栏目《面向方面编程、劫持和Unity 2.0》一文中，作者通过探究如何处理代码中的横切关注点，介绍了面向方面编程范例和Unity 2.0中的相关功能。

预测



@reichental

发布了针对CIO的2011年五大行业预测：http://oreil.ly/hIqw6I #CIO。



@Dzone

2011年最受欢迎的18种编程语言，http://dzone.com/JNSU。



@mashbusiness

2011年电子广告6大预测，http://on.mash.to/e8tmQS。



@LouisColumbus

2011年ERP产业预测，http://shar.es/XAtQe。



@chrisbrogan

2011年5种小型商业模式预测- http://ow.ly/3sPke。

资源



@DesignerDepot

做出好网站的20种设计教程（2010年度最佳），http://ow.ly/3sgB6。



@ftofficer

memcached的安全问题，sensepost在BlackHat USA2010上的主题，http://www.sensepost.com/blog/4873.html。



@daodao

够快：基于多终端支持的“个人云存储”服务（含公测体验邀请），http://bit.ly/f6ks5v。



@mujiang

Database Programmer 通过Tom的推荐又发现了一处数据库应用设计与开发的瑰宝级博客，喷血推荐，http://bit.ly/h3pBF0。



@radar

想知道#HTML5特性在不同浏览器上的工作情况？考虑一下这个兼容性列表，http://bit.ly/fx5JPf。



@kejunz

不得不赞一下facebook的这条设计原则：“reduce, reuse, don’t redesign”， http://www.lukew.com/ff/entry.asp?854。

文章



@bluedavy

【转】从多核到众核处理器by@biglizheng，这文章写的太好了，http://goo.gl/jiZ3C。



@cnbuff410

电脑帮助社交爱好者大开眼界，http://nyti.ms/f3AfZX。



@Carnage4Life

创业过山车，一个创业企业家的五步走，http://bit.ly/92ehxo。



@tony_yet

Steve Wozniak接受FCC的采访：坚持网络自由，http://j.mp/fyLcq6。



@andrewchen

Twitter正在发生怎样的改变：揭示Twitter发展方向的新研究，http://bit.ly/gqULVF。



@dalmaer

跨浏览器脚本加载效率的复杂性分析，http://ajxn.it/eriYKR，by@slicknet



@billwagner

博文：我对未来互联网应用的观点，http://bit.ly/gWmU0B。



@acumon

RT @ideawu：这篇文章值得一看：我希望NoSQL去死吧，http://is.gd/iOUQj。

（网址链接请访问杂志官网，www.programmer.com.cn）

CSDN最受关注文章 TOP10

（10月26日~11月26日）

1. 外媒评2010年科技业奇闻轶事：盖茨桌上跳舞
2. 2011 IT大趋势
3. Facebook vs.Twitter，谁更凶猛？
4. 博文推荐：腾讯这个公司
5. CA CTO：我犯过的最大技术错误
6. 两大国产操作系统联手抗微软：争夺“桌面”
7. 偷了世界的程序员
8. Tumblr的2010年有多成功？
9. 研究人员发布IE攻击代码
10. Facebook创始人低调访华，拜访百度

JavaEye最受关注文章 TOP10

（10月26日~11月26日）

1. Apache宣布退出JCP，包括 Doug Lea 和 Tim Peierls两位大牛！
2. 不适合当一名开发人员的10种迹象
3. 2011年Java EE生产力报告
4. 编程语言不需要开源
5. 最智能的Java IDE——IntelliJ IDEA 10发布
6. Oracle 发布 MySQL 5.5 正式版
7. 21个最佳jQuery插件推荐
8. Google开源WindowBuilder和CodePro Profiler
9. Google Wave现在是一个Apache项目
10. Vmail，Gmail 的vim界面

DCI 和 MVC

主持人：
潘加宇，UMLChina首席专家，潜心研究和实践UML／UP相关技术的应用。



软件工程与管理

2010年12月15日，Trygve Reenskaug给中国的开发人员作了题为“Models and Reality: Master Program Execution with DCI”的网络语音讲座。Trygve Reenskaug是挪威奥斯陆大学信息技术系的教授，从事编程和软件工程研究工作已经有50年。他率先研究的工作包括：面向数据库的架构、结构化设计和终端用户编程、面向对象应用和角色建模、模型－视图－控制器（MVC）范型、OOram方法及工具。其中最有名的就是MVC架构模式。该模式从Xerox PARC的Smalltalk开始，被实现在许多开发框架中，特别是在Web开发中大行其道。

而Trygve Reenskaug现在认为：面向对象一方面是程序的代码结构，由类及其继承结构组成，在编译时被冻结；另一方面是程序的运行时结构，由快速变化的对象通信网络组成。这两个结构基本上是独立的，难以通过研究代码来理解运行时的情况。程序员不能推断运行的正确性，被迫依赖于测试。数据－上下文－交互（DCI）范型通过架起编译时和运行时的桥梁弥补了这个缺陷。

Jdon创始人彭晨阳近年一直在研究领域驱动设计和DCI的进展，本文特地邀请他来发表针对DCI的看法。

彭晨阳：需求分析与代码设计统一新方法

DCI是一种构建面向对象应用的新途径。对象有数据属性和方法行为，以前我们是封装在一个对象中，因为这个对象在某个需求用例场景中被使用时需要这些属性和方法行为。注意，这里面有一个关键点，就是对象被使用。以前我们进行面向对象设计，是遵循一种静态原则，因为这个对象被使用需要这些属性和行为，所以，我们在编码时将这些属性和行为写在这个类中。

这个逻辑过程是不对的，那是因为过去程序语言平台落后，导致了我们这种思维逻辑。现在的思维逻辑是：对象被使用时需要的属性和行为不一定要在编写代码时写入，而是在运行时再注入或Mixin混合进去。这就是DCI架构的本质。

正如Rebecca Wirfs-Brock在其著作《对象设计：角色、责任和协作》中所提到的，MVC的控制器有很多缺点：控制逻辑会变得超级复杂，很多人在Struts的action控制器中写业务代码，所有的操作都在action中，有的action都几乎上千行了。控制器做很多事情，就意味着领域对象做很少事情，控制器最后不是只会做什么、决定战略性的事情，它也和怎么做、如何实现等战术问题耦合。

直接通过场景和视图绑定，比MVC模式中的Control要自然得多，这种角度更符合OO。MVC模式其实是一个结构性模式，因此不适合互动行为能力很强的应用，比如Web 2.0社会媒体或游戏等。所以，可以认为DCI架构颠覆了MVC模式。用DCI替代MVC，用场景替代控制器应该是大势所趋。

DCI的思想和Peter Coad的四色原型很相似。四色原型是需求层面，而DCI架构是技术实现层面。这两者类似，说明我们终于找到一种方法实现需求分析和设计代码的统一了。

特邀嘉宾：
彭晨阳（banq），Jdon创始人，上海解道计算机技术有限公司首席架构师、高级咨询顾问，OO软件思想布道者。



本月热点

新品发布

MagicDraw 17.0发布

新版本的改进体现在性能上的提升，加载模型的速度加快，内存的占用也减少了；另外还提供了Diagram Overview，以更好地展示和浏览模型。

针对Rails的UML工具

开源项目RailRoady，可以从Rails 3项目生成UML图。

资源

1979年Trygve Reenskaug提出MVC模式的文章

http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html

2010年12月Trygve Reenskaug关于DCI的演讲幻灯

http://heim.ifi.uio.no/~trygver/2010/China2010.ppt

2010年12月Trygve Reenskaug和中国开发人员交流录音下载

http://www.umlchina.com/Chat/Reenskaug2010.htm

中文资料

《DCI架构：面向对象编程的新构想》

此文作者Trygve Reenskaug和James O. Coplien，《程序员》杂志曾在2009年6月和7月分两期刊登。



主持人：
王翔，软件架构师，主要研究方向为XML、.NET、领域设计和PKI应用。工作之余喜爱旅游、写作和烹饪。

知识产权和数据库厂商博弈术

数据库

本月热点

●新品发布

甲骨文发布面向Oracle 11g R2（包括RAC功能）的新虚拟机模板

QLLabs发布支持Windows和Mac系统的多数据库转换工具SQLLiteConverter

新工具可将MySQL、PostgreSQL、Oracle转换为SQLLite。

http://www.sqlabs.com/sqliteconverter.php

●事件

SAP被判向甲骨文赔偿13亿美元

纽约时报（网络版）

http://www.nytimes.com/2010/11/24/business/24oracle.html

●推荐资源

NoSQL市场持续做大，至2015年预期规模将达到18亿美元

24-7pressrelease.com

临近年底，IT业除了要迎接各种“年度奖励”外，更要面临很多事情的“清算”。

2010年，来自知识产权的压力尤其大，中美贸易谈判的一个重点就是知识产权。区别于以往大概评估的清算方式，通过近几年在产品中“预留”的许可检查功能，今年微软、甲骨文、IBM等厂商更容易给中国的大型企业、机构开出清单，确认软件许可的购买缺口。尽管单纯从数量看，Windows、Office占比大，但即便超量也可以更多归为“个人”的“临时性”试用行为，但数据库软件的超量使用很难推卸为“个人”行为。DB2更多随IBM的主机搭售，而SQL Server和Oracle适用面更广、用户群更大，因此它俩自然成了这次清算的关键。

与此同时，位于大西洋两岸的两家长期竞争对手间也在进行类似的博弈。2010年11月23日美国联邦法院的陪审团最终裁定SAP向甲骨文赔偿13亿美元，用于结束这场历时3年的官司。13亿美元的赔偿也成为该国历史上最大的侵权赔偿费用， 而此前SAP已经支付了近1.2亿美元的诉讼费，如按照这个金额赔付相当于SAP要把去年大半年的利润转给甲骨文。这场官司起源于2007年，SAP收购的TomorrowNow公司非法下载了甲骨文大量的客户软件及相关文档，据称该公司用这些资料从甲骨文客户那边“挖墙脚”，而这些资料之前甲骨文是通过密码保存在它的官网。SAP的首席执行官Bill McDermott出庭时对非法使用软件（及附属文档）认罪，但并不认同因为资料流失导致甲骨文客户流失的说法，他认为SAP通过更好的折扣吸引用户。此外，为了明确侵权责任，SAP称TomorrowNow下载的这些软件只是它自己在使用，SAP母公司并没有访问。如果把这段IT八卦再往前点看，2005年，甲骨文收购了仁科公司（PeopleSoft），作为回击SAP马上就高调宣布收购了TomorrowNow公司（被收购前该公司是为甲骨文公司的PeopleSoft产品提供着第三方维护支持的一家公司），并为NetWeaver软件提供针对仁科软件的连接器，力争对甲骨文的收购案“釜底抽薪”。但TomorrowNow公司2007年的剽窃行为被甲骨文抓了个正着，随后甲骨文律师团队的诉讼也令SAP焦头烂额，SAP不得不为为子公司的不当行为承担责任，但相信后面甲骨文不会给SAP缓和的机会。

对每个企业尤其是大型企业而言，如果不能够对包括数据库软件在内的关键IT资产进行管理的话，就很可能被IT巨头们“点穴”，切断其赖以运营的IT系统，届时这些企业也许将为信息化的野蛮生长付出惨痛代价。数据库零散建设、分散管理的做法是我国很多大型机构前面30年政务信息化、企业信息化落下的病根，应用更新换代容易，但各核心系统生产数据库的整合一直都是老大难问题，如果不改变，未来最直接面临的就将是知识产权问题。看着焦头烂额的SAP，我们也许该行动了。

姗姗而来的国产企业开源产品

主持人：
邢波涛，现任北京新软孚信息技术有限公司技术负责人。关注SaaS管理软件和B2B、B2C电子商务的融合。新一代电子商务和SaaS管理软件积极的实践者。



企业应用与行业软件

相信大家对开源软件都已经不陌生了。开源产品也早已经深入到企业IT内部。从最流行的操作系统Linux、FreeBSD、OpenBSD，再到最流行的数据库MySQL，以及最流行的开发工具Eclipse，最流行的Web服务器Apache、最流行的Servlet容器Tomcat、最流行的企业开发开源框架Spring、Struts、Hibernate，统统都是开源的。即使到了云计算，像最流行的分布式文件系统Hadoop、最流行的NoSQL数据库Cassandra和MongoDB，最流行的搜索工具Lucene，也都是开源的。

不过，这些最流行的开源产品，统统来自于国外，反观国内企业，不要说影响巨大的知名的开源产品，即使是非知名的局部有影响的开源产品，出产的也不多。但是我在网上无聊寻找技术文档的过程当中，还真的发现了几款国产面向企业应用颇具潜力的开源产品。

让我为大家细数几款国内企业自己做的开源产品。首推当然是淘宝。2010年6月30日淘宝网开源平台已经正式上线。首批推出的项目包括淘宝开发的分布式KV存储及高性能缓存系统——Tair，同年9月还推出了TFS的开源项目，TFS是一个高可扩展、高可用、高性能、面向互联网服务的分布式文件系统，其设计目标是支持海量的非结构化数据。淘宝还开发了自己的Web开源框架Kissy，非常性感的一个名字。

第二个要推的就是人人网。虽然真假开心网大战，并没有想象的那么激烈和好玩，但是人人网却真的是在真刀实枪地做一个自己的技术。它不仅像淘宝一样弄出了自己的面向开发人员的人人网开放平台，还强劲地推出了自己的NoSQL数据库Nuclear。在2010年12月召开的第八届中国软件技术大会上，几乎每家企业都在宣称自己拥有了自主知识产权的云计算产品，不知道人人网是否申请了国家创新基金，如果没有，强烈建议申请。附带说一下的是，不知道是官方还是个人的，人人网有个研究员，弄了一个五四陈科学院，在那个网站上，他还推荐了疱丁分词作者王志亮在人人网的ose框架，使用上手快，配置少，并称其是创业公司Java必备。感兴趣的，倒也可以去尝试。当然了，只要是自己能做的，本人是不会使用外人的任何开源框架的。

第三个要说的就是天涯网。本人几乎不上天涯网的，但也知道天涯网流量大。天涯社区最近出人意料的开发了一款数据引擎——Memlink，并将其开源。Memlink是一个高性能、持久化、分布式的Key—>List/Queue数据引擎。正如名称中的Mem所示，所有数据都建构在内存中，保证了系统的高性能，同时使用块链进行内存压缩，使用redo-log技术保证数据的持久化。此外，Memlink还支持主从复制、读写分离、数据项过滤操作等功能。在我印象中，还以为天涯就是弄了一个类似Discuz的BBS论坛系统，看来天涯的水还真的很深啊。

本月热点

● 新品发布

Spring Mobile 1.0.0.M2发布

Spring Mobile是Spring MVC的扩展，用来简化手机上的Web应用开发。我真服了Spring这个东东了。跟任何流行的东西都有整合，无论是传统的S（Struts）、S（Spring）、H（Hibernate），还是Flex/ActionScript开发，或是Google的GWT和基于云计算的平台GAE，到处都可以看到Spring的身影。什么流行，就整合什么，唯恐自己跟不上形势，而落后下来，被别人不用。这种策略跟腾讯的QQ有一拼啊。

● 事件

Red Hat宣布收购云计算软件提供商Makara

企业开源软件厂商Red Hat宣布，已收购云计算软件提供商Makara。但没有透露具体条款。Red Hat计划把Makara的技术作为自己的“云计算基础”产品组合的一部分。

● 会议技术

2010年12月10—11日中国软件技术大会（第八届）在北京国家会议中心隆重召开

在此次大会上，几乎参会的每家企业都推出了具有自主知识产权的云计算产品和解决方案，真能够与时俱进的啊。



主持人：
姚磊，Microsoft Dynamics CRM MCP，多家企业信息化商业解决方案项目经验。熟悉IT规划与需求工程与项目管理。

电子商务公司与CRM

企业应用与行业软件

本月热点

●新品发布

嵌入式关系数据库ITTIA DB SQL发布3.3版本，提供图形化的开发、编辑工具

Unify嵌入式数据库SQLBase发布11.6版本，增强了监控和分析工具

面向Sybase ASE用户，IBM发布专门的DB2 SQL Skin，便于用户的T-SQL应用向DB2迁移

亚马逊Web Service提供关系数据服务的“读复制”

●事件

IDC发布关于Oracle Exadata Dabase Machine的用户评估报告

Marketwatch.com

PostgreSQL 发布针对一系列漏洞的安全补丁

http://www.postgresql.org/about/news.1244

●技术会议

Idera推出面向DBA的“SQL Server的秘密”系列在线教学视频prweb.com

http://www.prweb.com/releases/2010/10/prweb4604464.htm

在全球经济一体化的今天，信息技术的快速发展极大地改变了固有的商业模式，在竞争越来越激烈的市场环境中，建立基于信息技术的商业运行模式——电子商务成为企业首选的商业模式。在新经济时代，企业间的竞争规则发生重大变革，谁能掌握客户的需求趋势，加强与客户的联系，有效挖掘和管理客户资源，谁就能获得市场竞争优势，在激烈的市场竞争中立于不败之地。

客户作为企业最重要的资源，无论是传统商业模式的企业还是电子商务模式的企业，都面临着如何发展新客户、保持老客户、提高客户满意度、增加客户价值等问题。

CRM作为一种旨在改善企业与客户之间关系的新型管理理念，必将成为企业经营管理的核心理念。所以，无论将电子商务理解为一种平台或一种商业模式，企业实施电子商务的过程中以CRM理念为指导理念，都能够保证企业在电子商务模式下，充分发挥电子商务的优势。

通过CRM软件与电子商务系统之间的有机整合，能够促进CRM系统和电子商务系统充分发挥各自优势，帮助企业建立良好的客户关系。CRM对电子商务公司来说主要可以带来以下几个方面的提高。

一、可以通过先进的信息技术实现统一客户资源的统一管理，其中包括客户基本信息、客户的交易信息、客户关注内容、历史点击查看的感兴趣产品。客户会员信息、积分信息等、客户历史沟通投诉信息等。

二、实现对客户信息的整合以后我们可以通过商业智能工具对客户的信息进行挖掘分析。企业可以制定合理的客户分析模型，将企业客户按照客户的价值，客户所在的生命周期阶段结合客户的信息，将客户分成不同的群体，以调整不同客户群体需要采取的客户策略。

三、企业对消费者需求的响应态度将最终决定其在市场上的成败得失。以“一对一营销”为基础的客户关系管理也成为领导全球经济潮流的力量。通过商业智能分析出来的客户群体，我们可以通过CRM对客户其进行一对一的营销，将个性化的营销信息通过电子商务网站、呼叫中心电话、QQ等多种渠道推向客户，客户做出销售行为后，我们可以对其反馈情况进行投资回报率的分析。以便于我们后续优化营销策略。

四、现有电子商务公司都是通过多种渠道与客户进行沟通的，主要的沟通渠道是电子商务网站和实体店。当客户的消费习惯决定，这些是完全不够的，据统计，主要还有有如下几种：呼叫中心电话、短信、邮件、淘宝旺旺、QQ等。

现在这些沟通渠道的信息是相对独立的，没有统一管理起来并与CRM集成使用，所以客户在消费沟通过程中体验非常差。采用统一的客户沟通中心对电商企业提高客户体验，满足客户满意度有着相当大的作用。

创新触手可及：打造企业级 OSGi 应用

主持人：
俞黎敏，IBM高级工程师，满江红开放技术研究组织核心成员。开源爱好者，在各大技术社区为推动开源和敏捷开发积极摇旗呐喊、加油！



Java

当看到苹果工程师用积木搭建起古董级计算机时，让人想起技术创新与动手实践是多么的重要。OSGi框架是一个构建于Java虚拟机之上的模块化动态架构，向企业应用程序开发人员提供四个关键特性：生命周期管理，支持应用程序安装、更新、启动、停止和卸载，而不需要重启系统；模块化，减少较大的整体类路径上的依赖项；捆绑包版本控制，同一代码的不同版本可以共存，并可用于不同应用程序；减少Java EE应用程序的使用，使得多个企业应用程序包含相同的Jar类库包。

十多年来，OSGi技术在复杂性、可扩展性、可维护性、模块化开发方面都提供了充分的解决方案。随着商业与开源的OSGi应用容器的不断成熟与引入，由OSGi Bundle构成的企业级Java应用程序现在可以被开发和部署到这些容器中了。在过去一年里，有很多关于OSGi企业级应用程序是否就绪的激烈辩论。OSGi是一种已证实的且经过测试的技术，所以从这个意义上来说，它已经就绪有一段时间了。缺少的是Java EE开发人员正确利用OSGi企业功能所需的工具和框架的规范与实现。

OSGi联盟的EEG（企业专家组）成立于2007年，并且制定了OSGi功能包参考实现规范。EEG的目的是查看最常用的Java EE技术，并构建使用部署到OSGi框架的应用程序绑定技术的标准方法。OSGi联盟旨在促进OSGi服务平台的广泛采用，确保通过网络交付和管理的应用程序和服务的互操作性。引进企业级OSGi应用程序编程模型的关键因素之一是确保它是整个行业认可的标准。这是OSGi联盟的企业专家组（EEG）的焦点。该企业专家组的专家由来自IBM、Oracle、SAP、VMware、RedHat、Progress等的代表组成。通过良好的设计实践和方法，可以利用标准Java EE技术来创建模块化、可扩展的Web应用程序。但是，分离公共模块以便在多个企业应用程序之间共享，或者同时使用多个公共库版本时，就会遇到真正的麻烦。而OSGi是多年以来一直在应用服务器和Eclipse平台中使用的Java模块化技术。允许模块化企业应用程序直接使用OSGi技术来极大地简化开发、组装和部署。OSGi的这些特性提供了一个基础设施，模块化设计不仅是一个最佳实践，而且是唯一的实践方法。

将企业应用迁移到OSGi上需要一定的代价，这里面模块化本身的代价占去较大的部分，无论使用的是OSGi还是其他模块化架构，对大型、复杂、纵横交错的企业应用进行模块化要付出更大的代价，然而不这么做，应用程序就会随着时间的推移变得更加复杂，维护代价也会增加。OSGi并非万灵丹，使用OSGi像其他很多框架一样并不意味着一定就会成功，还经常需要在使用上与代码上进行一些变化以便进行迁移。新项目中引入OSGi比较容易，然而对于旧项目进行OSGi模块化改造则需要进行技术与代价中进行权衡。

本月热点

新品发布

JOpenID 1.08版本发布

OpenID是一个去中心化的网上身份认证系统。对于支持OpenID的网站，用户不需要记住像用户名和密码这样的传统验证标记。相反，他们只需要预先在一个作为OpenID身份提供者的网站上注册。JOpenID是一个遵循OpenID 2.0规范的非常快速的轻量级Java客户端，仅50KB+，允许任何Web网站通过OpenID支持用户直接登录而无需注册，例如Google Account或Yahoo Account。JOpenID支持Java 5或更高版本，Servlet 2.3或更高版本，能运行在任何Java Web应用中。在新发布的JOpenId 1.08版本中，新增了对Maven的支持。

Spring LDAP 1.3.1发布

Spring LDAP是一个简化LDAP操作的Java类库，吸取了Spring JDBC的JdbcTemplate模式而创建。通过LdapTemplate类，开发者可以不需要考虑烦琐的DirContext的创建、NamingEnumeration的循环遍历、大量的异常处理以及相关资源的清理；从而让开发者更专注于从哪里找查找数据，以及如何去处理这些数据。此外，Spring LDAP还提供事务支持、池类库、Object-Directory Mapping（ODM）框架以及与Spring Batch集成的LDIF解析类库。

最智能的Java IDE——IntelliJ IDEA 10正式版发布

JetBrains出品的IntelliJ IDEA是一个屡获殊荣的Java和Groovy集成开发环境，也被认为是当前Java开发效率最快的IDE工具。它整合了开发过程中实用的众多功能，几乎可以不用鼠标可以迅速地完成你要做的任何事情。



主持人：
高昂，中国标准化研究院助理研究员，从事信息技术标准化研究工作。关注开源社区，也是OSGeo中国和InfoQ中文站成员。

DOI：国际化的数字对象标识符

编程

本月热点

●新品发布

Mozilla开发的编程语言——Rust

Mozilla目前正在开发名为Rust的编程语言，由Mozilla的CTO同时也是JavaScript之父Brendan Eich主持。Rust语言吸收了动态语言的特性，主要目的是为了帮助计算机软件实现在语言级别上真正利用多核计算带来的性能提升。

●事件

WindowBuilder和CodePro Profiler开放源代码

Instantiations公司的专注于提供基于Eclipse的Java开发工具，日前基于Eclipse的Java GUI设计器WindowBuilder开放源代码，WindowsBuilder提供了基于Swing和SWT的设计器，帮助开发者简化界面的定制工作。同时，另一款基于Eclipse的应用程序内存泄露和性能分析工具CodePro Profiler也将开放源代码，贡献给Eclipse基金会。

●资源

中文DOI：http://www.chinadoi.cn/

中文DOI是国际DOI基金会（IDF）授权的DOI注册机构，提供中文数字资源的DOI注册与解析服务，实现中文数字资源相互链接，与国外出版机构使用相同的标识与链接标准，实现中文数字资源与西文数字资源的链接与共享，方便用户获得中文数字资源。

ISO近期以100%的赞成票投票通过了一项新的国际标准“ISO 26324：数字对象唯一标识符（Digital Object Unique Identifier）”，简称DOI标识。DOI由统一资源定位符URL演变而来，为网络环境下的信息资源实体确定唯一的标识规则。

ISO国际标准化组织是成立于1947年2月的全球性非政府组织。ISO标准的制修订工作由各技术委员会负责，ISO/TC46第46技术委员会是信息与文献技术委员会，负责制定和推广与信息和文献工作相关的国际标准，DOI标准由ISO/TC46的分技术委员会SC9归口管理。除DOI标准外，技术委员会还负责国际标准书号ISBN、国际连续出版物号ISSN和国际标准音像号ISAN等标准的制定和修订工作。

在信息化时代，数据和信息本身的价值往往比作为载体的管理信息系统更重要。在DOI出现前，大量有价值的数据因知识产权得不到保护而不愿共享。此外，很多具有研究和使用价值的数据由于缺乏统一标识规范，常会造成无序的引用和歧义。针对上述问题，数字对象唯一标识DOI通过构建数据资源标识方法，为数据提供注册、解析和查询途径，在推动数据资源共享的同时为数据的网络传播提供版权信息。

DOI标识编码由前缀和后缀两部分组成，中间用“/”分割。为满足可扩展的数据资源编码需求，DOI没有限制前缀和后缀的字符长度。DOI前缀由目录代码和登记机构代码两部分组成，所有DOI代码目录定义为“10.”，即DOI 代码均以“10.”开头。登记机构代码通过向国际DOI基金会（IDF）申请获得。登记机构代码的分配根据信息资源编码需求制定，如出版商可为其所有数字出版物申请统一前缀，也可以为数字图书、音像制品等各类出版物申请不同前缀。DOI后缀是在特定前缀下的唯一标识，由登记机构分配并确保其唯一性。后缀可以是任何字母或数字编码，如机器码、ISBN 或ISSN等已有规范码，具体编码方案由登记机构设定。根据编码规则，DOI数据资源命名规范通常定义为：“统一前缀/类型标识.机构名.内部标识码.扩展码”的形式。

目前国外Elsevier、Blackwell、Springer、John Wiley等大型数字资源提供商已开始使用DOI对其出版的电子文献和数字内容进行标引，帮助基础性数据资源的快捷获取和灵活引用，同时为了便于数字资源的共享与交换，DOI登记机构还会在唯一编码基础上提供数据资源的命名、注册、解析、查询和变更等管理功能。DOI能够实现资源实体的永久性标识，便于数字资源的归档与检索。目前专业的DOI检索站点CrossRef已与Google搜索技术结合，在DOI检索过程中提供数字文档的全文检索，为用户建立检索结果到数字文献全文间的获取通道。

DOI标识的唯一性、持久性、兼容性、互操作性、动态更新等特点，必将会在信息资源标识引用过程中起到越来越重要的基础性作用。

让“云”落地



主持人：
涂曙光，微软平台技术专家，专注于.NET、Web、SharePoint、Office等技术领域。现任职于中国惠普有限公司。

.NET

只要你是一名经常关注微软技术社区以及微软技术动向的开发人员，一定已经知道云计算已经成为微软的重要战略方向。无论是美国的PDC 2010技术大会，还是12月份在北京举行的TechED China 2010技术大会，你能看到的几乎都是有关云计算的内容和讲座。这一切会让所有开发人员都多少有些心存疑惑：“云计算真的已经成熟了吗？”

从技术上来说，笔者认为各个厂商的云计算平台，具体到微软也就是Windows Azure平台，已经基本进入了一个相对的稳定期，其功能、开发模型、开发工具已不再发生大的变化。虽然12月份微软又发布了最新的Windows Azure SDK 1.3版本，但也只是向Windows Azure增加了一些增强的功能和更多的Virtual Machine模板。

对微软而言，下一步更重要的可能是考虑如何进一步提高开发人员和技术社区对Windows Azure的接受度，吸引更多的开发人员为Windows Azure创建真正有用的云应用，打造更多的成功案例。这样才能形成良好的正向循环，将越来越多客户吸引到Windows Azure上来。Windows Azure在当前最大的阻碍，并不是Google App Engine、Amazon EC2这些云计算领域的竞争对手，而是客户与开发人员对云计算的担心和犹疑。

当然，如果说到微软的云计算，仅仅说Windows Azure这个PaaS平台，未免也不太公平。因为除了Windows Azure之外，微软今年刚刚发布了一个新的SaaS云计算服务：Office 365。Office 365是原来Business Productivity Online Services的升级版本，它将SharePoint Online、Exchange Online、Lync Online和Office Pro Plus打包成标准的服务，允许企业客户按照所需的时间和用户数量，向微软订购这些在线云计算服务。相对于Windows Azure，笔者倒是认为Office 365对企业客户有更大的吸引力。任何一家企业都需要邮件服务以及安装在员工电脑上的Office软件。对于大一些的企业，可能还会需要诸如IM服务、在线会议服务、在线协作服务，等等。这些服务实际上非常适合于以“标准化”方式提供给企业。大部分企业对自己的邮件系统都不会有什么特殊的自定义需求，与之相对的则是对邮件服务的可靠性有很高的要求。将这些“标准化”服务以云计算方式进行采购，免除自己采购服务器、部署系统、管理运维的负担，对企业客户有很大吸引力。

当然，如果整个.NET平台只有微软一家提供的Windows Azure云计算PaaS平台，未免太冷清了一点。幸好社区的力量是无穷的，一家名为Sine Signal的公司，就宣布将基于Mono构建一个云计算平台Moncai，然后对公众提供云计算服务。可惜现在Moncai还处在初始阶段，不过你可以到moncai.com网站先注册一个Beta版邀请。笔者希望未来能出现更多提供.NET/Mono支持的云计算平台出现，让.NET开发人员拥有更多的选择。

本月热点

● 新品发布

Visual Studio 2010 SP1 Beta

Visual Studio 2010发布了第一个Service Pack的测试版，如果你是一个喜欢尝鲜的开发人员，可以通过下面的网址下载它：http://go.microsoft.com/fwlink/?LinkId=207140。

● 中文资料

● 技术活动：Silverlight Firestarter 2010

Scott Guthrie在2010年12月2日主持了Silverlight Firestarter 2010技术活动，在这次活动中，Scott Guthrie和其他来自Silverlight团队的工程师展示和介绍了Silverlight 5，并讲述了其他一些高级主题。整个技术活动的视频资料可以通过此网站在线观看：http://channel9.msdn.com/Series/Silverlight-Firestarter。

● 培训课程：Unified Communications “14” Labs

Microsoft Lync Server 2010和Exchange Server 2010都为开发人员提供了更丰富的开发接口，大大扩展了其开发扩展性。这个培训课程就是针对这两个产品，讲述了如何通过它们的接口对其进行扩展开发。课程的在线观看地址：http://channel9.msdn.com/Series/Unified-Communications-14-Labs。

主持人：
钱宏武，现任职盛大创新院，原搜狐互动产品开发部主管，资深互联网社区架构师，垂直搜索领域专家。



从很傻很天真到维基解密

Web

本月热点

事件

Linux 再爆 root 账号提权漏洞

程序利用了 Linux Econet协议的3个安全漏洞，可以导致本地账号对系统进行拒绝服务或特权提升，也就是说一个普通用户可以通过运行这段程序后轻松获得Root Shell。

评价：补上漏洞，看清楚人才是最重要的。

OpenBSD开发者否认加入FBI 后门

Gregory Perry声称FBI付费给OpenBSD的IPSEC协议栈开发商NETSEC，让他们在代码中植入后门。现在原开发者已经正式发表回应，要求Gregory Perry公开道歉，否认植入了任何后门。

评价：传说很多操作系统都有后门，但就像鬼一样，无数人都听说，但谁都没见过。

外媒评选过去25年10大黑客攻击事件

美国新闻网站The Daily Beast评选出过去25年中10大最具破坏性的黑客攻击、蠕虫攻击以及分布式拒绝服务攻击事件。

评价：看看吧，网络世界中，小偷、强盗、流氓还是蛮多了，因此才有这样的榜单。

当年很傻很天真的照片外泄，是由于使用的人不了解计算机的原理，从而让维修的人轻而易举的把那些图片搞出来，并且放在网络上。但对于维基解密来说，大概谁也无法知道信息是怎么得到的。而他们的网站，也碰到了信息安全问题：它的DDOS攻击。2010年12月3日，EveryDNS宣布，由于WikiLeaks.org在公布美国国务院保密文件后，遭遇大规模的DDoS攻击，导致流量剧增，被迫停止向该网站提供域名解析服务，以免其他50万个使用EveryDNS服务的网站受到影响。

DNS的攻击对于我们来说，大概已经很常见了，这个互联网的基础服务是如此的重要，却又如此的脆弱，熊猫烧个香就能让无数人无法上网，一个免费软件的问题就能搞崩溃几个省的网络。今天让美国政府颜面扫地的维基解密也要面对这个问题。这种基础服务的升级看来是必然的。

这次WikiLeaks受攻击是否为美国政府所为已经不重要了。和一般公司束手无策相比，我们看看这些Geek们如何应对EveryDNS的行为吧。维基解密已在全球多个国家建立了超过1000个镜像站点，为了防止DNS提供商停止提供域名解析服务，维基解密与包括瑞士、加拿大和马来西亚在内的8个国家的不同DNS提供商签署服务合同。从这个我们能够看到，对于一个网站来说，如果说这种攻击是无法避免的话，那么唯一能做的，还是传统的智慧，不把所有的鸡蛋放在一个篮子里。如果内容重要，尽可能的备份你所有的内容放在尽可能多的地方。

2010年底美国新闻网站The Daily Beast评选出了过去25年中10大最具破坏性的黑客攻击事件，也让我们看的大开眼界，在这次评选的10个事件，一个很有意思的共同点，全部是以一人力量去对抗一个巨大的组织、国家、公司、组织，甚至对抗全球，可见技术这种事情，不是说你人多，就有优势。想起一句话：下围棋，找100个业余选手一起，也无法打败一个一流高手。

由于安全的原因，很多人纠结于用什么操作系统。近期系统安全高手Dan Rosenberg 发布了一段 C 程序，这段200多行的程序利用了Linux Econet协议的3个安全漏洞，可以导致本地账号对系统进行拒绝服务或特权提升，也就是说一个普通用户可以通过运行这段程序后轻松获得Root Shell。

其实这个安全问题不是很可怕的漏洞，因为要能利用这个漏洞，就必须能够登录服务器。如果出了问题，只能说是管理问题，为什么会让一个不可靠的人登陆你宝贵的服务器？而很多服务器出问题，很多黑客事件，事后被证明很多并不是黑客有多高明，只是管理者的疏忽，如弱口令，或者密码有太多人知道，或者服务器的安全漏洞没有注意打上补丁，而如果安全管理能够很全面，我想谁都不会很容易地成为是很傻很天真的主角。

游戏人遭遇寒冬



主持人：
房燕良，普通程序员成长起来的资深游戏制作人，投身游戏业10年以来从未脱离开发一线。主要代表作品有《仙剑奇侠传3》、《功夫世界》等。

游戏开发

前几天，北京连续大风降温，据说有可能遭遇千年一遇的寒冬。而这个冬天对于一些游戏人来讲，确实遭遇了前所未有的寒冬。年关将近，游戏公司裁员的新闻频频报出，包括九城、久游、金山、完美、联游在内的多家公司报出“人员调整”、裁员等消息。据消息称，近一个月被裁员或者被待岗的约有800人左右。这和年初业内纷乱的挖角、跳槽现象形成了鲜明的对比。

前几年，国内的游戏产业发展经过几个投资的高潮，已经经过了相当长时间的爆发式发展。对一个行业来说，能够如此快速地发展十年，已经实属不易。很多公司也经历了飞速的扩展过程，在进入平稳增长期的时候，进行一些调整也是有必要的。我们看到像完美这样强大的公司，也进行了“人员调整”。但这在企业发展的过程中也是正常的，有利于企业的健康发展。对于整体的游戏行业来说，也需要挤挤泡沫啦！据业内人士称，近期不单单是大公司裁员，还有不少团队或小型公司被撤资或解散。经过2010年二季度的首次负增长，以及后面的增速减缓，网游的爆发增长期肯定是已经过去了，将进入平稳增长期。很多投资人本身就是看中“爆发式增长”而来的，看到这种势头，选择离开也是属于正常的投资行为。这些资金的离去对行业来说并不是坏事。

很多人都感慨游戏圈的浮躁。正是前几年的爆发式增长，一个个成功的神话，确实对我们这些从业人员心理有非常大的影响。很多人通过一个成功的项目就可以身家几百万，甚至很多人项目没成功，混了几年资历，薪资也能翻几倍，这使得很多从业人员也是急功近利。

被裁员或者被投资人抛弃的团队，最需要的是好好调整自己的心态。首先是要借这个机会好好地思考一下自己的职业生涯如何发展，抛弃所有急功近利的想法，必须要能够沉下心来做事。其次，要好好思考为什么自己被裁掉？前几天和一位不幸被裁员的朋友聊天，这位仁兄可以说火气很大，不断抱怨公司管理的问题，项目管理的问题，团队管理的问题等等，就是没有谈及自己的问题所在。一个项目被裁员，肯定是有很复杂的原因，不过，我希望他还是能够先从反思自己的问题开始，把所有的问题都推给别人、推给环境，只能使自己变成一个“怨妇”。

在我们能够静下心来，反思自己的不足，好好思考自己出路的时候，大家也不必悲观。只要是真正有实力，又能够踏踏实实做事的人，在游戏业还是有很多机会的，毕竟游戏产业还是要继续发展的。

对于需要找工作的兄弟姐妹，个人有一点建议：要好好挑选公司，最好是进入还处在增长期的公司。最关键的是一定要参与一个能够成功的项目，并踏踏实实做好自己的工作。最后祝愿所有同行朋友们过个好年，来年事业更上一层楼！

本月热点

新品发布

中娱在线《猎刃》

2010年11月17日，北京中娱在线网络科技有限公司对外宣布，Q3已成功融资2亿元人民币。同日，中娱旗下的3D动作网游《猎刃》也开启了开放性测试。在游戏中玩家将扮演猎人的角色，挑战各种强悍的远古怪兽，游戏强调动作设计和打击感。

事件

《星际2》拟明年初大陆上市

2010年11月18日消息，网易相关人员表示，已经向相关政府部门提交了《星际争霸2》的信息和资料，正在等待审核阶段。另外，据知情人士介绍，暴雪近期已经在筹备《星际争霸2》中文版上市发布会事宜，但是具体时间暴雪也未确定。

搜狐畅游首次代理韩国网游

搜狐畅游于2010年11月18日举行的韩国Gstar游戏展会上正式宣布代理韩国网游《KARCASSOnline》。该产品是由韩国JCRSoft公司研发的动作角色扮演网游，以15世纪中期欧洲骑士反抗恶魔侵袭为背景。

中文资料

首届中国原创网页游戏博览会在成都举行

2010年11月20日，首届“中国原创网页游戏可持续发展高峰论坛暨中国原创网页游戏博览会”在成都开幕。此次盛会由文化部、成都经信委、成都博览局指导，成都天府软件园主办，共有约100余家国内外知名的网页游戏开发商及运营商参加。

推荐资源

Code Sampler网站

http://www.codesampler.com/

主持人：
马宁，微软最有价值专家，Windows Mobile开发者。



Chrome OS和云操作系统

新平台

本月热点

●新品发布

2010年12月7日，Google发布Android 2.3和Nexus S手机

Android 2.3 Gingerbread终于在万众期待中闪亮登场。Android 2.3进行了界面优化、提升运行速度、提高Open GLES和Open SLES等游戏支持功能。除此之外，Android 2.3还优化了电源管理、多媒体文件支持、应用程序管理和下载管理器。最吸引眼球的功能是增加了NFC Reader，NFC是一种近距离通讯技术，是RFID的升级。

与Android 2.3同时发布的还有Nexus S手机，Nexus S的制造商是三星，Google试图打造最优的Android手机，没有搭载任何第三方应用。集成NFC Reader，是Nexus S手机另一大看点。

Google开始发放NFC标签

在将NFC技术作为Android 2.3的标配后，Google在波特兰开始为商家发放NFC标签。NFC技术由飞利浦和索尼联合研发，将RFID读卡器和智能卡结合在一起。在日本，很多手机应用NFC技术进行手机支付相关的业务。而Google将NFC应用在LBS服务中，最近推出的Google Hotspot服务，只要让Android手机刷一下商家的NFC标签，就可以方便地看到与该商家相关的评价信息。

2010年12月15日魅族发布M9手机

经历了无数次跳票，魅族M9手机终于正式发布。告别了Windows CE，M9采用了UI深度定制的Android系统，相对于缺少第三方应用的M8，M9无疑会得到Android开发者的追捧。M9采用了三星S5PC110处理器，主频1G，960x640分辨率显示屏。虽然M9外观设计摆脱了iPhone的影子，但硬件配置还是贴身紧逼iPhone 4。在中国，肯花几年时间打造一个产品的公司不多，魅族无疑是一个理想主义者的公司。祝魅族好运！

2010年12月14日，我们终于看到了Google Chrome OS的庐山真面目，这款以云技术为核心的操作系统，同时发布的还有类似工程样机的上网本CR-48。虽然外界对于Chrome OS的前途有诸多疑问和批评，但Chrome OS毕竟代表了未来的一个发展方向——云操作系统，尽管大方向如此，但Chrome OS到底是先烈，还是先驱，就要看Chrome OS是否能够满足目前的用户需求。

先来看看Chrome OS有什么吧，CPU采用Atom处理器，所有数据存储在云端，Shell与Chrome浏览器紧密集成，基于HTML5的离线功能，允许用户在离线状态下对Google Document进行编辑，Chrome Web Store支持程序下载，提供免费和收费的应用程序；支持3G和WiFi网络。除了玩游戏外，Chrome OS该有的功能都有了，不过，Chrome浏览器已经开始支持WebGL了，所以玩游戏的问题会随着系统成熟逐步被解决。

不过Chrome OS的成熟度的确很难达到PC操作系统的要求，Windows在PC操作系统领域屹立多年，其雄厚的技术实力，也不是能够在短时间内颠覆的。最简单的例子，Windows目前支持数万种硬件驱动程序，可以保证Windows正常运行在x86的硬件平台上。所有驱动程序必须通过微软的LTK测试，从而保证Windows系统的稳定性。Chrome OS目前只支持定制的上网本硬件，如何说服硬件厂商为Chrome OS/Linux开发驱动，是Chrome OS推广上的一大难题。

Chrome OS也不是完全没有机会，比如一些企业及应用领域，为了便于管理，将所有数据都存放到云端，可以有效地保证数据安全。不过Google在企业级应用领域没有一套完整的解决方案，这一点会制约Chrome OS的推广。毕竟使用Windows Embedded的瘦客户端解决方案已经非常成熟了。

Chrome OS其实是云操作系统的代表之一。云操作系统的概念，最早是由一家叫做Good OS LLC的公司创造的，他们推出了一款基于Linux的gOS。该操作系统基于浏览器，集成了Wine、Picasa和Google Gadgets等应用和服务。

除了gOS之外，还有几个类似的解决方案，比如Webconverger，集成了Firefox和Flash Player等。虽然云操作系统大都采用Linux作为底层OS，但微软也发布了自己的云操作系统——Gazelle，允许Web程序直接在安全内核中执行，任何代码产生的崩溃都不会影响内核的运行。

云操作系统的概念其实被很多云计算服务所使用，比如Live Mesh中的虚拟桌面、腾讯的WebQQ，虽然只是一个基于Web的虚拟桌面，但假以时日，这些技术也完全有可能变成云操作系统。

目前看来，Chrome OS是成熟度最高的云操作系统，但对于游戏和硬件支持的欠缺，可能很难在短时间内得到根本改观。不过，云计算都来了，云操作系统还会远吗？

移动平台新版本纷纷发布



主持人：
崔海斌，十几年的开发经验，专注于Java、Android、移动开发等领域。现就职于创新工场旗下的点心团队就任总架构师。

移动

Android的新版本2.3（代号Gingerbread）终于于12月17日发布在AOSP上了，第一版是2.3.1，而之前的SDK早在12月6日就已经发布了。新版本确实如之前Google承诺的有多项改进，尤其对开发者来说，增加了很多比较好的Debug的API，如StrictMode，通过此API，开发者可以在自己应用的调试中监控到如磁盘读写、网络使用、设置有非法操作或者故障产生时的反应，甚至可以直接同步到Dropbox内，再直接在编程用机上打开此日志，而不是之前只能看到FC和ANR的提示窗口。而对于普通用户来说，最大的变化一个是整体界面的风格有所变化，之前的白底黑字的Status Bar和Notification以及Menu都变成了黑底白字风格，还有一个相当重要的功能改进就是选择模式下，增加了选择区域微调的功能，虽然没有iOS那样的放大镜微调效果更佳体贴，但是对于绝大部分用户来说已经足够好了。此外，系统中增加了VOIP/SIP协议的支持，据说还可以实现视频通话。而在其他开发人员密切相关的方面，不得不提的是Google最近对Market升级，而分级制度的引入，Widgets和动态壁纸专用的动态分类的增加，更多的介绍图片，版本release的change log，对于设备硬件的要求，上传APK大小限制从25MB提高到50MB，这都是好消息。而最重要的更新是退款时间从原来的24小时缩短到了15分钟，相信可以让那些付费应用的开发者受益匪浅，原来那些24小时内打通关就退款的用户将无法再利用这个免费玩游戏了。

随着iOS新版本4.2的发布，Apple宣布所有运行iOS4.2的iPhone设备将会得到免费提供的手机找回服务，这确实对于iPhone的用户是一个的利好消息。另有传言说明年2月份左右就会有iPad2出货，而首批出货量为40万~60万台，目前无法证实其消息真伪。

微软官方博客最近发布，WP7上的第一个主要更新版本（代号Mango）将于2011年8月或9月发布，而第二个更新版本（代号Apollo）将会在2012年底之前才会发布。而之前传言的2011年2月的更新只会包括多任务解决方案、增强的开发者控制、应用内下载以及其他一些定制化的选项的一个“小”更新。在笔者和一些业内人士看来，这更像一个“主要”更新。根据WP7上的最近应用数量统计，WP7 Marketplace上的应用数量已经突破4000个应用，虽然比起App Store和Google Market差距很大，但也是相当不错的成绩了，有个对比就是Palm应用商店在2009年6月开张，到目前为止，上面的应用数量只有大约5800个左右。

RIM在12月16日公布的第三季度的营收报告，称其第三季度营收和出货量都比去年同期增长了40%，这个报告超出了大部分机构对其的预判。不过比起其主要对手Apple和Android来说，实在属于一个很难拿出手的数字。对于四季度的预期，RIM的发言人特意提到了对PlayBook的预期。

本月热点

● 新品发布

三季度Apple的iTunes音乐商店保持增长

根据最近NPD集团的一个市场调查报告显示，三季度Apple的iTunes音乐商店占据了66.2%的在线音乐购买市场份额，比去年同期增长了63.2%。这是否会成为继Appstore以后又一个大家纷纷进入的市场？不过相信至少在中国大陆短期内不会出现。

RIM收购TAT

2010年12月2日，RIM宣布收购了著名的设计团队TAT。TAT在设计界可是鼎鼎大名，号称设计界的SWAT，希望能够尽快在未来的PlayBook和黑莓手机中看到其设计产品。

Android系统占智能手机市场一般份额

根据Morgan Keegan对于中国市场的3季度统计显示，3季度中国大陆共销售800万~1000万台智能手机，而去年同期只有200万~300万台，其中基于Android或兼容的系统占总销量的接近50%，而运行iOS系统的只有不到50万台。此报告不知是否包括水货市场，不过情况应该也是大致差不多，当然，如果包括iPad和iTunes，此数据应该会有些变化。

Gallery3D更新

Android 2.3 Gingerbread中的Gallery3D包含众多的更新，修复了众多Bug。我原本以为2.2版本开始会转入闭源的Gallery3D居然更新了，看来Google没有完全放弃开源的Gallery3D分支。

主持人：

郑叶飞（怿飞/圆心），淘宝前端技术专家，D2前端技术论坛发起人之一，专注前端技术，关注用户体验，乐于前端布道。



前端，继续前行

Web前端

本月热点

●新品发布

Google 正式发布Chrome OS

Chrome OS就是Web操作系统的概念，大部分的程序与资料都是在网络上处理，运行在浏览器上，“理念”比较超前，也许未来会给前端带来更好的发展。

Mozilla宣布计划2011年初发布跨浏览器应用商店

2010年12月6日，在Google宣布Chrome Web Store应用商店发布之后，Mozilla也表示计划发布跨浏览器的Web应用商店。App主要基于HTML、CSS、JavaScript等技术，可以运行于PC、移动等终端，支持目前的多数高级浏览器。

●事件

Web Developers & the Open Web 2010

由Mozilla发起的一项调查，就Web开发者的职业类型、技能语言、JavaScript框架、技术社区等方面做了详细的了解，并以此为基础，将MDN建设得更适合开发者。

微软高管称中国是IE6消亡最大障碍

很多原因导致中国用户一直在使用IE6，而这一现状使得Web新技术的推广举步维艰。推动IE6消亡的工作，需要整个行业的同仁一起联合起来，加速IE6退出历史舞台的进程。

●技术会议

2010年12月8日，Velocity China 2010

2010年Velocity大会——Web性能和运维大会在北京成功举办，此次会议邀请了众多国内外嘉宾，分享Web性能、基础架构、稳定性等方面的经验，受到了国内众多大公司和开发者的高度关注。

转眼间，前端在国内已经走过整整五年，回顾五年的发展，令人欣慰：国内各大公司开始重视这个职位，前端的从业人员不断增多，前端职位的需求量不断膨胀。

什么是前端

个人理解的前端就是从数据到展现，即：后台开发提供前端所需的数据，前端拿到数据，通过简单分析、解析，甚至再包装，通过（类）浏览器展现给用户。

前端的技能成长

前端要将页面在（类）浏览器中展现给用户，就需要用HTML来写页面的结构，用CSS写页面的表现，用JavaScript写页面的交互行为，也就是说前端必须掌握HTML、CSS和JavaScript，至于程度至少要能够运用它们完成大多数任务。前端还需要处理后台开发封装的数据，处理方式有两种：一、使用Ajax的技术，通过异步处理数据；二、使用类PHP的脚本语言，在模板中处理数据。至于第二种方式很多人可能有疑问，这部分不是后台开发做的事情吗？之所以放在前端，是因为前端更懂用户体验，更懂结构（HTML）和行为（JavaScript），更懂业务，前端做这部分工作更有理由，而且后端可以抽出更多的时间，处理后端的性能、算法和逻辑等。所以，个人认为前端必须掌握 HTML、CSS、JavaScript、（类）PHP技能。除这些基本技能之外，还应掌握可用性、可访问性、前端测试、前端安全、前端性能等扩展技能。

前端未来的“战场”

掌握了各种前端技能，我们可以看看前端未来可以做些什么？

HTML5的出现推动了前端的快速发展，你可以用Canvas实现图表和游戏，你可以使用 WebSocket实现与服务器端的实时交互，你可以在本地存储和读取数据，你还可以非常简单地实现动画。

很多网站都提供了API接口，我们可以基于某些网站的API接口，开发出新奇的应用、甚至有价值的站点。比如我们使用Google Map的API接口，就可以开发出公交路线查询的功能。

我们可以轻松地开发Chrome、Firefox等浏览器的插件，只要你会JavaScript，就可以创造奇迹，你的未来或许离不开Google的Chrome Web Store 和 Mozilla 的 Open Web App Ecosystem。也许未来某一天，就可以用JavaScript来编写OS应用。

NodeJS的出现，赋予了前端更广阔的发展前景，技术人员可以像写JavaScript一样，来写服务器应用，一劳永逸，无需学习后端语言，前端将更自由。

前端的未来，依旧很光明，当然需要我们学习的知识也很多。

（声明：以上观点，仅代表个人的看法和想法，不代表任何公司和组织。）

开源、开放与安全



主持人：
肖新光，网名江海客，安天实验室首席技术架构师，研究方向为反病毒和计算机犯罪取证等。

安全

2010年12月，金山网络宣布部分产品开源，并按照预订的计划路线图逐步公开了漏洞修复等功能模块的代码、公开了云查杀接口。

Windows平台软件，特别是安全软件，选择开源的相对较少，这可能与Windows本身的封闭性有关。尽管有开源的呼吁，但微软目前也只是进行针对一些国家政府机构的代码托管，相关代码是不含加密认证等模块的不可编译代码。站在公共信任的角度，源码开放是一种公众审核机制，而代码托管则是第三方审核机制。因此，相比于Windows的代码托管策略，开放源码的Linux在开放性上获得了更好的口碑和认同。

国内安全界对开源的感情十分复杂。在中国信息安全企业起步的过程中，对IPtable、Snort、Nessus、FreeS/WAN等开源软件的学习和借鉴无疑降低了技术门槛；但技术门槛下降导致人人皆可为之而又不思进取的小厂商林立，把汉化包装成自主原创、同时又不遵守开源许可协议的情况十分普遍。

当然开源的问题也是明显的，从安全风险来看，二进制可执行文件面临的风险主要来自逆向分析、错误注入和一些条件构造，但源代码开放后则面临可能被攻击者进行源码级漏洞挖掘的风险。

此外，病毒检测的机理如果被公开，就会有更多病毒制造者学会绕过反病毒机制，或许这也就是金山对云查杀只开放了接口的原因。

但开源如果为主流厂商所引导，就会有很大不同。金山开源应该是在当前国内反病毒桌面竞争的一个胜负手，其对产业的深远影响还需要进一步观察和评估。但可以肯定，对于网络安全爱好者和相关专业的学生，他们有了更多可以参考的范本。

同时，也需要提醒安全领域专家学者们，今后在国家科研项目审查中，即使是Windows平台，也要防备有人盗用开源代码应付了事——过去，专家们揪出了很多的Snort马甲、ClamAV马甲，现在来看还会出现金山马甲。最后，还希望金山意识到开源不只是厂商开放代码给大家用，能形成一个对代码开放、共建、共享的体制、让更多人交互的参与，才是开源的精髓。

美国国家标准技术研究协会（NIST）对下一代散算法SHA-3的开放式甄选已经到达最终冲刺阶段。从2008年64个算法参选、51个进入第一轮开始，经过长达两年的两轮甄选后，本月有5个算法进入了最终的决赛名单。它们是BLAKE（瑞士）、Grøstl（Graz理工大学和丹麦理工大学合作）、JH（伍宏军，新加坡华裔学者）、Keccak（Belgian Daemen团队）和Skein（Bruce Schneier团队）。

2012年，我们将知道SHA-3花落谁家。值得回味的是，MD6在去年退出了，被称为优雅的转身。从DES到AES、从SHA-1到SHA-3，美国这种开放的态度、制定规则的先见，从容地使全世界数学精英为其打工。

本月热点

新漏洞

国家信息安全漏洞共享平台（CNVD）发布了48期漏洞通报

通报中共收集、整理信息安全漏洞103个，其中高危漏洞29个、中危漏洞22个、低危漏洞52个。上述漏洞中，可利用来实施远程攻击的漏洞有93个。经跟踪发现，网上已经出现针对“Enano CMS SQL注入漏洞”、“Microsoft Internet Explorer CSS解析远程拒绝服务漏洞”、“Freefloat FTP Server ‘USER’命令远程缓冲区溢出漏洞”等的零日攻击代码。

详见：http://www.cnvd.org.cn

新闻

关于SHA-3算法遴选更多信息可以参见：

http://www.h-online.com/security/news/item/NIST-s-search-for-the-super-hash-just-five-candidates-left-in-SHA-3-final-1151325.html

新品发布

安天发布免费数据恢复工具

http://www.antiyfx.com/a-recovery.html



技术社区的新发展

——记2010 SAP TechEd

■ 记者 / 董世晓

技术社区以分享知识和体验为目的，社区成员在这里可以通过论坛、博客和维基百科等平台，进行广泛的交流与协作，以便他们能通过协作创新来解决技术和业务上的难题，并得到快速、高质量的响应。作为一种传播先进技术、展示公司形象的途径，技术社区目前已经为各大IT公司所广泛使用。

而随着社交网络和成员需求的发展，技术社区也呈现出了多种多样的状态。SAP创新社区就是比较有代表性的一种。SAP创新社区建立的初衷，是为了满足其客户不同层面的企业管理应用需求，打造以客户价值为中心的生态系统，从建立至今，经历了SAP开发者网络（SDN）到后来的业务流程专家社区（BPX）、大学联盟社区（UAC）、企业服务社区（ESC）、行业价值网络（IVN）、合作伙伴资源中心（EcoHub）、博客等。而在刚刚结束的2010 SAP TechEd上，我们获悉了SAP创新社区的新特点。

与第三方社交媒体全方位合作

社交网络的出现，降低了人们社交的时间和物质成本。国外如Twitter、Facebook、YouTube、LinkedIn和Flickr，国内如新浪微博、开心网、土豆网等社交媒体目前已经深入人心，人们的生活方式和习惯也随之发生了明显的改变。技术人员作为新鲜事物的践行者，社交媒体对他们的影响更为深刻。

正是基于对这些新形势的捕捉和理解，SAP创新社区在第三方社交网站上设立了专区，从而使社区成员与SAP专家在上面进行方便的沟通。据SAP创新社区全球高级副总裁尤腾（Mark Yolton）先生介绍，SAP创新社区目前有来自200多个国家和地区的200余万名成员，中国则有5万多位，而且都已经实现了通过第三方社交媒体的协作和交流。

搭建线下人际关系网络

长期以来，SAP一直致力于为社区成员搭建完善的线下人际关系网络。在本届TechEd上，随着云计算（Cloud）、移动应用（Mobility）、基于内存的计算（In-memory Computing）几大技术发展方向的盛大发布，SAP又适时地推出了几种新的线下交流形式，帮助SAP技术拥趸进行更多的思想碰撞和经验分享。

创新演示擂台赛（Demo Jam）和创新社区客堂间（Mini Clubhouse）的引入，受到了参会者的广泛关注和一致好评。创新演示擂台赛是SAP提供的一个展示自我及企业创新力的舞台，今年是首次登陆中国。通过一系列角逐，入围SAP大赛的作品亮相擂台赛，由现场观众直接选出最佳人气奖。创新社区客堂间的增设则提供了更多互动交流的渠道。社区客堂间除准备了CRM快速部署解决方案、源代码交换平台等精彩讲座外，参会者还与擂台赛参赛选手、贤明顾问、社区活跃贡献者探讨了如何更好地在创新社区通力合作从而实现共赢共生，了解SAP生态系统如何为知识整合创造条件，并即时分享在大会上学到的新知识，针对SAP的最新技术趋势和发展动向各抒己见。

此外，为了给TechEd参会者交流讨论提供更多的便利，SAP在现场还细心摆放了技术话题贴纸，特别加长了课间休息的时间等。

可以说，SAP创新社区的建设，是技术社区发展的一个缩影，其采取的多种新形式，可以为其他技术社区的发展提供借鉴和参考。



SAP创新擂台赛颁奖仪式欢欣鼓舞

TechED China 2010，“云”的大会

■ 记者 /佳琪

2010年12月1-3日，5000名IT技术人汇聚北京，参加了微软TechED China 2010技术大会。如果要对TechED China 2010做一个总结，那就是：一次“云”大会。从大会最开始的Keynote，到各个技术课程，都充满了有关云计算的内容，TechED China 2010可以算是微软在国内进行过的一次最大规模的云计算技术宣讲活动了。

云适合谁

在大会主题演讲中，微软大中华区董事长兼首席执行官梁念坚表示，云计算是一个非常重要的概念，无论从技术方面还是IT发展方式都是非常有利的。他认为中国的云计算发展潜力很大，尤其是对中小企业。因为云计算这种方式把整个IT行业做了改变，而且门槛降得很低。梁念坚还表示微软云计算将会促进中国企业的发展，特别是中小企业的发展，而且是配合中国政策，同时推进中国十二五计划。大会上微软联合用友畅捷通、SAP、HP和ESRI等多家云计算合作伙伴，全面了展示“云+端”战略和解决方案。

云中微软

TechED China 2010举办期间，Windows Azure也发布了其最新版本的开发工具与SDK v1.3。1.3版本的Windows Azure增加了一些新功能，例如对运行的Virtual Machine的管理员权限和远程桌面连接、新的Virtual Machine版本等。

对此，梁念坚也鼓励VMware用户尝试使用Windows Azure云计算平台，为企业提供新的云体验，也给微软一次机会。他表示，“做云计算开发肯定有钱途，微软也会持续不断地对云计算投入大量资源”。



梁念坚表示云计算就是一个大家分享的资源

云里云外

除了云以外，本次大会还有相当丰富的产品发布与推介，但我们从中仍能窥到云的影子。诸如，微软发布的Lync Server 2010通过和Office 2010、SharePoint 2010、Exchange 2010，特别是微软云办公软件Office 365的整合，完成了在云时代下统一通讯领域的布局。

作为大会亮点的Windows Phone 7在第一天的讲演中重磅出场，整合性的服务、便捷快速的系统及软件开发环境是Windows Phone 7区别于其它系统的明显标志。我们可以说Window Phone 7是为云而生，其开发平台均可以在微软云中得以应用，并且在消费领域承担由“云”到移动“端”的连接。

浏览器作为互联网的入口，与“云”更是密不可分，此前，微软首席执行官史蒂夫鲍尔默就曾表示微软的IE9将提供多个客户端满足用户对云计算服务的需求。

在大会IE9 Beta版的介绍中，笔者了解到了包括硬件加速、像运行桌面软件那样运行互联网应用、将互联网应用”固定”在任务栏上的应用与概念，IE9在微软“云”战略中举足轻重的地位可见一斑。

纵览本次TechED China 2010大会，下面的四字横幅充分表达了微软的总结和展望：会——此次大会，动——云计算已在行动；耘——一起努力耕耘；酝——终将酝酿成霖，普惠大众。

三江汇聚，历久弥新

——记甲骨文全球大会

■ 记者 / 佳琪

2010年12月13-15日在北京举行的“甲骨文全球大会”，有个重要变化——实现了“三江汇聚”，即首次在同一城市、同一周举办了甲骨文全球大会、JavaOne和甲骨文开发者大会。尽管会议的组织形式有所变化，但甲骨文一贯对技术和创新的执著，还是一脉相承，并且历久弥新。在本次大会上，Oracle分享了他们多项最新技术和产品，包括MySQL 5.5、Java SE/EE/FX/ME、Glassfish和NetBeans、SPARC超级集群、中间件云服务器系统、第四代超级计算机、应用开发框架移动11g、Cloud Office 1.0、Open Solaris 11 Express等。

Web为王的MySQL

自Sun被Oracle收购之后，开发者对MySQL的前途难免会有一些担心。虽然此前，Oracle的首席架构师Edward Screven曾对社区表示，Oracle不但会继续对MySQL企业版做出贡献，也会继续大力支持在GPL协议下社区版本的发展。并且Oracle对MySQL的投资也会比Sun多很多，并让MySQL适合大企业的需要。但是直到大会上Oracle才外宣布，MySQL 5.5全面上市，才让人对MySQL的未来有了切实的希望。MySQL 5.5的功能此次得到显著增强，可以在Windows、Linux、Oracle Solaris、Mac OS X等多种操作环节下运行，并改进了Web应用的性能和可扩展性。另外一点比较重大的变化就是社区版按照GNU通用公共许可证（GPL）的规定提供许可，包括默认存储引擎InnoDB在内，可供用户免费下载。在Oracle提供的官方新闻稿中，提到了采用MySQL 5.5版本可以让用户受益的几个地方：



甲骨文产品开发副总裁Amit Zavery阐明Oracle对云计算的全面支持

- 改进的性能和可扩展性：MySQL数据库和InnoDB存储引擎已得到更新，在运行于最新的多CPU和多核硬件及操作系统上时，提供最佳性能和可扩展性。此外，在5.5版中，InnoDB是MySQL数据库的默认存储引擎，可提供ACID交易、参照完整性和应急恢复。
- 更高的可用性：新的半同步复制和复制“心跳”（Replication Heart Beat）功能改进了失效转移速度和可靠性。
- 改进的可用性：改进的索引和表划分、SIGNAL/RESIGNAL支持以及包括新的PERFORMANCE_SCHEMA在内的增强诊断功能改进了MySQL 5.5的可管理性。

根据在开源基准测试工具Sysbench所做的测试，新版的MySQL在性能上也得到提升。在Windows上运行时，MySQL 5.5 RC版和MySQL 5.1相比，读/写操作时性能提升15倍，只是读操作时性能提升5倍。在Linux上运行时，读/写操作时性能提升3.6倍，只是读操作时性能提升2倍。

针对MySQL 5.5的发布，“甲骨文公司首席设计师Edward Screven表示：“最新版MySQL进一步表明，甲骨文兑现了对MySQL社区的承诺，正在不断投资以尽快为MySQL平台增加创新和增强功能。有了MySQL 5.5，最终用户和独立软件开发商（ISV）在开发和部署关键业务

Web应用及嵌入式应用时，就有了一个可以替代Microsoft SQL Server的高性能、可靠、可扩展和经济实惠的产品。”

Java命运备受关注

在JavaOne大会上，Oracle提到了他们在Java SE、Java EE、Java FX、Java ME等领域的进展。开发人员比较关注的Java SE 7将会在2011年夏天就绪，其中JCP已经批准了四个JSR（334、JSR 335、JSR 336和JSR 337），支持改进Java语言和API规范，比如switch in string、multi-catch、泛型实例创建中的precise rethrow、Lambda表达式（闭包）等。

Java SE 8也将在SE 7发布之后的一年到一年半的时间就绪。JavaFX将和Java整合，并运行于JVM，其富媒体处理及二维、三维等渲染功能会提供给任何在JVM上运行的编程语言，比如Groovy、Scala和JRuby等，其中Java X UI控件将被开源。

因为收购Sun公司，Oracle拥有了两个主流的JVM（ HotSpot和BEA的JRockit），在这次大会上，它们将如何整合进Oracle的产品线也有了明确的战略：在Oracle的规划中，两种Java虚拟机的资源将会合并到一个团队/代码库中，比如将JRocikit的任务控制台移植到HotSpot JVM等，并将整合的JVM通过OpenJDK开发源代码。

应用服务器GlassFish将通过加入几个WebLogic的功能进行强化，并发布开源和有Oracle支持的付费等两个版本，在Oracle的方案中，GlassFish继续为其J2EE实现的首选工具。

NetBeans也将在2011年带来新的版本，包括发布开源和收费两个版本，前者为社区版，不能享受来自Oracle的服务支持。

Oracle这次还推出了基于SPARC T3和M5000服务器的SPARC超级集群，这是一个较为全面的基础设施解决方案，可以用来运行Oracle数据库真正应用集群（RAC）环境。



甲骨文产品市场营销集团副总裁Robert Shimp表示云计算是大势所趋

SPARC超级集群解决方案的架构与创造了新的TPC-C基准测试世界纪录的Oracle系统所用架构完全相同，该解决方案还采用了新的SPARC服务器、FlashFire、InfiniBand QDR、Oracle Solaris和ZFS存储设备。该超级集群将于明年上市。

全面提供云计算

在中间件领域，Oracle通过其融合应用软件推广其“应用无极限”的下一代应用软件理念。简单来说，Oracle融合应用软件作为一个平台，结合来自Oracle客户的最佳实践，基于标准的中间件基础、采用服务导向方法和公用数据模型等开发，并可以部署在私有云、公有云等上面，让用户既可以继续现有的管理软件应用，也可以采用共存并行的策略，或者使用完整的套件，通过不同途径实现价值最大化。

虽然从前拉里•埃里森对云计算有过相当刻薄的评价，但在云计算一波波的浪潮冲击下，Oracle也不得不顺应大势。

这次的大会上，Oracle不仅宣布在北京成立甲骨文云计算解决方案中心，还阐述了甲骨文云计算策略：云意味着数据中心架构的根本转变，是将软件作为服务提供的基本方法，甲骨文的云计算最大的目标是确保云计算完全达到企业级，以便企业能安心采用。此外也适时推出自己的云计算产品：Oracle数据库云服务器（Oracle Exadata Database Machine）和Oracle中间件云服务器（Oracle Exalogic Elastic Cloud）等。

其他看点和热点

本次甲骨文全球大会另外一个看点，就是韩国科学与技术信息研究所基于Oracle产品成功部署了第四代超级计算机，位列全球最大超级计算机TOP500名单上第24位。

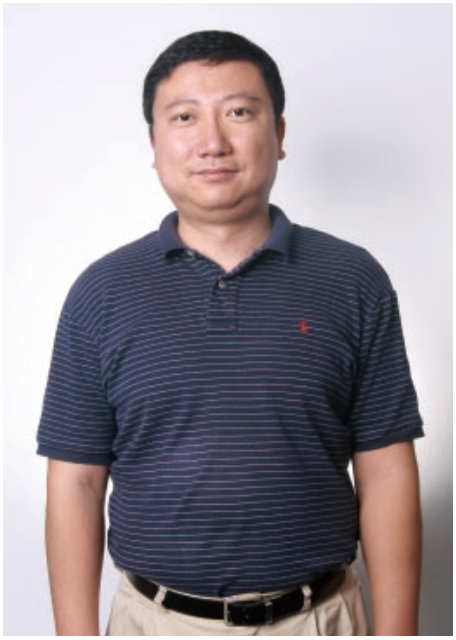
这个超级计算机的核心是Oracle Sun Blade服务器以及Oracle Sun Infiniband、Oracle Sun Lustre文件系统和Oracle网格引擎（Oracle Grid Engine）。该超级计算机的组件之间是通过Oracle Sun Datacenter Infiniband Switch 648和Sun Datacenter Infiniband Switch 36连接的。

在目前的另外一个热点技术移动开发领域，Oracle推出的应用开发框架移动11g帮助韩国综合性金融服务集团实现了移动办公环境，Java ME将来会包含WebKit引擎、JavaScript引擎等，这也意味着未来运行新版Java ME的手机，将可以访问浏览器以外的Web服务，而不存在任何性能问题。

信任的力量

——走进NHN 中国技术开发中心

■ 记者 / 常政



崔海表示，NHN中国技术开发中心的目标是打造一支拥有全球化的开发竞争力的软件研发团队

NHN，是韩国最大的互联网公司，拥有世界排名前五的搜索门户网站Naver和全球最大网络游戏门户网站之一Hangame。很多中国IT人士知道NHN，是因为它曾经在2004年购买过联众的巨额股份，但并不为业界所熟知的是，其实自2008年起，NHN还在中国成立了研发中心，如今已经拥有400多名雇员，分布在北京、延吉、沈阳、青岛四地。NHN能成为韩国搜索和网游的翘楚，相信国内的开发者们一定会感到好奇：他们的技术理念、工作模式有哪些是值得我们去学习、借鉴的？为此，《程序员》记者来到了NHN中国技术开发中心。

据NHN中国技术开发中心总监崔海介绍，NHN在中国成立开发中心的初衷，是看中了中国丰富的技术人才资源，便把公司内部的产品分包到中国来做。当时还不是以技术开发为主，但随着NHN的进一步发展，尤其2009年，NHN技术团队开始革新自己的研发模式，为实现更高效的开发效率，NHN逐渐转变了中国开发中心的定位，使其专注于技术本身的研发工作，成为NHN重要的技术和人才储备基地。崔海说：“NHN中国技术开发中心现在的目标是打造一支拥有全球化的开发竞争力的软件研发团队。”

NHN的技术支撑

作为NHN的技术支撑，崔海团队致力研发的核心产品也和NHN的核心业务密切相关。目前NHN旗下的Naver.com，搜索页面流量超过63亿的PV，并拥有超过170万的独立访问者；而它的游戏门户Hangame峰值在线人数为29万。无疑这需要一个稳健高效的计算平台。经过一番摸索，NHN推出了面向服务的平台搭建——基于PaaS(Platform As A Service)的计算平台，同时这个平台中的许多核心模块均由中国开发中心来实现。据NHN Middleware平台开发业务组长孙国伟介绍：“这个平台分3个层面，最底层包括海量的文件系统、海量的数据管理系统。在中间主要有三块：消息平台，系统模块和公用模块。消息平台用于容错，系统模块针对应用系统开发者，公用模块针对服务商。最上层是服务运行环境。”除此之外，NHN针对这个平台还提供了开发者环境和开发工具。

在NHN看来，随着互联网网站的暴发性增长，服务平台越来越不堪重负，开发、维护费用也呈指数级地增长；而PaaS，通过实现平台计算的服务化，可大大提高网站的运营效率。

开源的影响力

除了计算平台，整个NHN产品服务的后台数据库CUBRID，也由中国

开发中心提供技术支撑。令记者感兴趣的是，CUBRID是一款开源产品。说到个中缘由，崔海解释有如下三点：

首先，对于互联网行业，目前鲜有针对中小型网站使用的、可支持大规模访问的数据库产品。所以我们把产品开源后，会吸引更多的网站在CUBRID上进行创新，而不必去购买非开源的商业数据库产品。

其次，开源不仅能扩大公司的行业影响力，而且能延续产品技术本身的生命力：一旦这款产品被广为接受，再借助社区型的社会协作开发，可以更加繁荣它的技术体系。同时为了更好的和社区开发者互动交流，我们还在CSDN网站建立了NHN中国开发中心团队博客http://blog.csdn.net/nhn_devlab。

此外，自实施开源后，出现了一始料未及的现象：公司技术人员的开发热情和效率显著上升。因为一旦开源，全世界都可能有你的用户（而不是公司内部几个人用），这大大提高了开发人员的成就感。

据NHN DBMS开发部Team长白桢焊介绍，CUBRID包含四个子项目：数据库引擎、数据接口（Interface）、数据管理工具、集群，其中后三项均由中国开发中心来负责。据统计，CUBRID目前下载量是每月1000次左右，用户主要来自韩国和欧洲。

崔海说：“开源同样给我们带来了产品经营理念的变化，原来我们主要考虑公司内部的具体应用需求，现在更多关注的是业界的应用标准和共性需求。”

信任的力量

除了技术支撑，中国开发中心同样是NHN的人才储备基地。如今外资软件企业，薪资福利相差无几，所以团队文化和成长环境将是吸引高素质人才的关键。崔海坦言，初到NHN时，颇为招聘高素质人才而苦恼过，由于大家对NHN不熟悉，所以开始时关于高端人才的招聘广告总是应者寥寥。但随着一年多的团队建设，NHN特有的企业文化、以创新为导向的工作模式开始在中国落地生根，渐渐在中国高端IT人才中辐射出自己的影响力，NHN终于构建出一支颇具竞争力的技术团队。

谈到心目中的理想人才，崔海说：“他有对技术的热爱、有勇气面对挑战；富于创新精神，总是不满足于现状，总是在搜寻更好的Idea或解决方法；以始终负责任的精神和谦逊的态度来工作，能有效地建立起员工间和团队间的互信和合作。”而这种对人才的要求，即“Innovation & Creatity;Passion & Challenge;Modesty & Sense of Ownership”构建出NHN中国开发中心特有的团队文化。

来自北大的应届毕业生范一璠（DBMS开发部、DBMS工具开发组初级软件工程师）对于NHN新员工的入职程序印象深刻，他说：“首先会安排一名老员工做你的导师，然后让你去熟悉、撰写代码模块并对它进行答辩。经历这些后，会接到一些诸如改Bug的简单任务。最后等你对这些很熟悉后，会被正式安排一些开发工作。”对于这样的设计，李成龙（DBMS开发部、DBMS开发组组长）解释这是为了了解新人的特长：“对于新人，最重要的是让他做自己能胜任的事情。所以我们不会给他很大的压力，我们看重的是培养他做事的正确方法和独立思考的良好习惯。”

于淼（DBMS开发部、DBMS开发组中级软件工程师）是从另一家外资软件企业跳槽而来，她说：“我觉得NHN的特点，就是无论遇到什么问题，都不用等待，你可以立即召集周围的人，去一个会议室讨论。”

在谈到对于团队氛围的最大感受时，无论是范一璠还是于淼，竟不约而同地表达了一种“被信任”的集体认同感。范一璠说：“一般应届毕业生到一家公司都会从事边角料的打杂工作。但我在NHN仅过半年的时间，就开始负责与CUBRID的Interface相关的核心工作，这对刚毕业的学生来说是一个非常难得的机会”。以上的访谈点滴，不难发现NHN团队文化的特色：宽松愉悦的工作氛围下，是理性、科学的做事原则，是对人才不问资历的择才量用，并使他们井然有序地各就其位。而员工们直观感受到的，又是一种集体归宿感。为此，白桢焊总结说，一支好的团队标准就是“创新，互相信任，凝聚力”。

技术开发全球化2.0

某种意义上说，一家技术企业的文化氛围和它采用的研发模式密不可分。白桢焊认为，



接受访谈的NHN中国技术开发中心团队成员（从左至右）：崔海，白桢焊，孙国伟，于淼，范一璠，李成龙

崔海领导NHN中国技术开发中心一年多的时间里，最大的贡献是使得整个团队从“技术开发全球化1.0”上升到“技术开发全球化2.0”：前者以技术人力资源为出发点，优势体现在开发成本上的竞争力，运营方式是由总部统一进行项目管理，各分部在项目层面进行合作化开发，但这种模式会增加一些额外的成本（比如沟通等）；而后者以技术开发的成熟度、竞争力为核心价值，优势是通过各部门、项目团队产品和技术的平行化开发，实现项目并行、产品互补，既可以提升产品竞争力，也有益于项目团队的管理。

所以，在这种更加关注“技术竞争力”的开发模式里，“人”的因素，尤其是“创新的自主性”被置于了主导地位。崔海认为NHN中国技术团队经过“技术开发全球化2.0”模式改造后，目前最大的特点是高效、灵活。崔海说：“比如在NHN，产品部门经理几乎可以全权决定所有相关的产品细节，而高层仅仅是做诸如‘做还是不做’这类宏观决策。”

结束语

在参观NHN 中国技术开发中心时，令我印象最深刻的是，NHN在办公区内自建了一个小型咖啡馆。时值上班时间，但还是看到一些年轻的开发者们在里面，要么独坐沉思、要么三三两两围坐私语，一副怡然自得的气象，使人很难想象这是一家商业公司。一个关键字不由得在我脑海浮现——信任。这种信任不仅体现在NHN对员工的态度，也贯通在其经营理念里，比如公开核心技术的开源模式，其实也体现了对产业和技术社区的信任。尽管业界对NHN总部的中国战略猜想还显得莫衷一是，但崔海和他的团队一定会稳健地走下去。因为如崔海所言，NHN总部同样也对中国开发中心寄托了无限的期望和信任。

Office系统对信息工作人员价值最大

——微软MVP 专访

■ 记者 / 杨东杰

2010年10月29~30日，微软中国在上海举办以“荣耀之光，王者同行”为主题的2010 GCR MVP Open Day，数百名两岸三地IT行业精英参加本次盛会，在会后，本刊记者特别采访了微软MVP周庆麟，周庆麟是Excel Home的创始人和站长，早在2002年就获得过微软MVP称号。

记者：你第九次当选微软MVP，最大的感触是什么？

周庆麟：从最初的Power User到技术专家，再到技术作者和培训讲师，MVP项目伴随并推动了我个人职业生涯中的每一步。在MVP项目中，我有幸与来自两岸三地的牛人们结识，有幸直接与微软产品工程师探讨产品特性，有幸远赴新加坡、美国等地参加技术研讨。最重要的是，MVP项目让我及时把握最新技术趋势，认识前进的方向，并不断鞭策自己提高。

作为IW（信息工作者）类别的MVP，我最初的专业其实是财务管理。工作四年后，出于对电脑技术的热爱和对Office软件的痴迷，我从财务工作转行到IT管理工作，重点关注领域是IT技术在中小企业中的应用，并曾作为IT经理主导并实施过多个企业信息化项目。在众多的微软产品和技术中，我认为对于广大信息工作者而言，最有价值的还是Office System。

记者：“信息工作者”指什么？为什么说Office System对他们来说最有价值？

周庆麟：这里的“信息工作者”指的是当前最广泛普通的一个IT技术使用者群体，他们在各自的专业领域中，需要按各种需求来处理数据和文字、管理信息。通常只接受过最基础的IT技术教育，没有时间和精力来关心任何深奥的IT系统和技术，但他们都渴望IT技术帮助他们进行高效工作。因此，一旦公司部署的IT管理系统无法完全满足业务需要，Office软件就能很好地弥补所有空隙，在信息管理的最后一公里帮助用户完成工作。甚至在很多情况下，灵活而功能强大的Office完全可以胜任多变的普通管理任务。

从另一方面说，每家公司可能部署着各种类型的IT管理系统，但他们大多数都会用到Office。因此，熟练掌握Office 已经成为信息工作者的一项基本职业技能，同时，所有投资在学习提高Office应用水平上的时间和金钱都将转化为长期有效的职业资本，不受工作调换所带来的影响。

记者：请谈一下你自己的技术成长经历和心得，以及你接触到的国内信息工作者对Office工具的使用情况。

周庆麟：我最初接触到Office，是在毕业后的第一家公司里，当时主要使用Excel 5.0来处理大量的财务数据，用Word 6.0制作财务报告。除了使用财务软件完成最基本的会计核算以外，几乎



周庆麟认为Office完全胜任多变的普通管理任务

所有的其他财务数据处理和报表计算都要用Excel完成。作为职场新人，我当时就意识到Office软件的重要性，并投入很多时间和精力来学习它，终于从最初的菜鸟变身为公司里的Excel大拿，最终在事业发展上获益匪浅。

和早年间相比，Office最主要的组件Excel和Word无论是功能数量还是易用程度都已经不可同日而语。遗憾的是，我还是看到绝大多数用户并没有真正充分利用好Office软件来解决日常业务问题。现在的Office组件众多和功能更加强大，但使用者的情况依然是一般水平多良好掌握少。实际上，日常工作中最常用的还是老四样：Outlook、Excel、Word和PowerPoint，大家可以结合自己的业务带着问题一步一步提升。

互联网不适合螺丝钉员工

编者按：互联网本身变幻万千，技术发展一日千里，猫扑网技术总监陈杰认为，如果你像一颗螺丝钉那样按部就班，并不能把自己的全部潜能都发挥出来，也难以将自己的智慧体现在产品上。互联网需要你有极客精神，在专注的领域做到极致。



陈杰 猫扑网技术总监

记者：请简要介绍一下您的成长与求学经历？

陈杰：我是在吉林大学读的计算机专业。记得2001年毕业前，我们毕业生被要求填一张叫做“职业理想”的表格。我写下的理想是“为中国软件事业做些贡献”。记得当时很多同学为图省事“抄袭”了我这句话。但后来我转移到了互联网，没有在软件领域做更多的深入。

记者：您为什么要转移到互联网领域呢？

陈杰：是中国软件业的行业状况促使了我的转变。中国软件企业主要做的是外包类项目，提供行业应用的解决方案。在这种企业里面，技术只是一颗螺丝钉，起主导作用的是行业里经验丰富的人，纯做技术的路很窄。而在国外，技术人员有很宽阔的发展路线，可以在技术领域钻研得很深。此外，国内很多领域的软件项目，均是利用国外的成型技术，比如数据库用Oracle的、中间件用IBM的。这样，利润最厚的部分都被国外企业挣了。国内企业利润薄，投入也会小，导致国内软件行业技术人员发展的路也就窄了很多。

互联网行业则截然不同。虽然互联网企业也常引用国外的技术，但它孕育了像腾讯、百度这些排名世界前列的企业，它们在大数据量处理、高并发访问上丝毫不弱于国外公司。这说明中国的互联网企业，已完全可以通过努力在国际市场上争取到更大的份额；此外，中国有全球最大的互联网用户群，所以当用户数量从几千万数量级增长到几亿数量级时，由此引起的成本增长会比较小，而相应带来的利润却非常丰厚。这是互联网的魅力所在，这样企业对技术的投入也会水涨船高，能给技术人员提供广阔的发展空间。

记者：猫扑网现在的技术重点是什么？您如何规划猫扑网未来的产品走向？

陈杰：目前，我们投入精力比较大的就是猫扑Hi，它比较像人人网的SNS，但又和它有着本质上的差异。人人网的SNS是一个采用真实姓名、真实身份的公共交友平台，而猫扑Hi则将自己定位为一个虚拟社区。这说明，千橡公司的整个产品构成是比较有趣的：猫扑网打造虚拟世界，人人网还原现实世界。

2009—2010年，我们一直在加大用户体验方面的投入，使得用户体验满意度不断增加。具体来说，就是对我们过去已有的产品进行精耕细做，详细研究每一个方面，尽力改进所有的不足。我们会分两部分来做：第一，尽可能提高底层逻辑处理速度；第二，尽可能提高页面展示的执行速度。

记者：您觉得一个合格的技术领导者应该怎样管理团队？

陈杰：千橡公司有它自己的公司文化，对于团队成员，我的要求是：首先，团队中的每个成员都要有强烈的责任感。互联网行业与软件行业有所不同：在软件行业，当一个团队领导者设计好产品的架构，每个成员就好像是一个机械上的螺丝钉，大家按照规程做就可以了；但互联网行业，如果你只是这样按部就班地做事，可能完成了领导安排的任务，但最后的产品未必能成功。互联网本身变幻万千，技术发展一日千里，如果没有强烈的责任感，就不会把自己的全部潜能都发挥出来，也就不会将自己的智慧体现在产品上。其次，我要求团队的成员有一个积极的态度。因为，每一个产品都是为了迎合某种需求而设计。现在的很多互联网产品都做得比较急，并没有很好地考虑到使用者的全部需求。所以如果每个人都能有积极的态度，发现问题、解决问题的时间就被会大大提前，这对产品的用户体验和迭代更新都是非常有意义的。（记者/张雪峰）

技术创新是从业准则

编者按：从中国到美国，她深深感受到两者在技术文化上的差异；从卡耐基–梅隆到Google，她体会到技术理念决定产品成败；从Google到MediaV（聚胜万合），她把技术创新作为自己的从业准则。

美国的导师都称为advisor

记者：从清华到卡耐基-梅隆，您觉得差异最大的地方是什么？在留学期间有什么难忘的故事？

胡宁：相对于国内的高校，美国的教育是一种多维度的素质教育。清华的学习生活，尽管相对于国内其他高校，已经算比较辛苦；但当我走进美国的大学，发现这里的学生其实更加辛苦。他们学习和作业的任务量比我们大很多，经常可以看到凌晨两三点钟还有学生在实验室里做实验。我看到国内很多报道称，美国的教育是“傻瓜教育”，学生的数学能力非常差——这是不真实的，会误导我国的学生。只要你真的进入美国的大学，就会发现他们实际上学习很努力，并且受到的教育也是全方位的，所以他们的综合素质很高。现在的美国学生读大学是很辛苦的一件事情，这与中国学生高考前发奋读书，高考后彻底放松的情形截然相反。

另外，我觉得美国院校的学术氛围非常浓厚，比如在我就读的卡耐基-梅隆大学的计算机科学专业里，导师都称为advisor。而在国内，很多导师其实叫boss（老板），这两个称呼的含义是截然不同的。

记者：我注意到您在卡耐基-梅隆除计算机科学硕士学位之外，还取得了娱乐技术硕士学位，能讲讲这方面的情况吗？

胡宁：娱乐技术是一门交叉学科，同学们都来自于不同的（本科）院系，比如计算机科学系、机械系、建筑系等。该专业旨在培养娱乐技术界的人才，比如游戏领域。这样的行业对人的素质要求很高，因为牵涉多个相关领域，需要综合性的素养，所以我们所学的学科范围非常广泛，比如设计、戏剧理论、音乐制作、计算机图像等，它们都与我们的专业有关。比如说，在我们的课程中会讲到如何设计UI、怎样的游戏情节会有足够的戏剧冲突等内容。所以我们学习这个专业的压力也很大。当时比较让我难忘的一件事情，就是卡内基-梅隆有一个著名的、由Randy Pausch主讲的课程，叫《Building Virtual Worlds》。这门课的结课作业是做一个虚拟世界，我当时带领一个小组来完成，自己主要负责的是一个网络协同模块。后来的结课表演，就是大家一起参加、一起互动。所以这样的教学方式，给我留下了深刻的印象。



胡宁 MediaV CTO

技术创新是从业准则

记者：您是怎么加入Google的？从读博士转到从事实际工程研发，最大的差别是什么？

胡宁：2004年我去Google实习，结束以后Google就向我发出了邀请。当时我还是有些犹豫，毕竟我们学校毕业的学生很容易找到助理教授的工作，从而可以比较容易地走进学术界。但加盟Google后，我发现自己真正喜欢上

了Google的氛围，同时也经历了从读博士到工程研发的转型。在Google做项目的过程中，我感受最深的是接触的数据量和以往大不相同。因为我在读博时所做的研究项目，能拿到的数据大概是在十万这个量级，但是加入Google后发现，一个很普通的项目，所处理的数据可能就是上千万、上亿的量级。这是一个非常有意思的变化，因为在数据量增长到一定规模以后，它体现出来的性质跟小数据量的性质存在很大的区别。

记者：我们知道您是谷歌音乐搜索项目的技术总监，能否讲讲这个产品研发过程中曾经遇到的困难？

胡宁：这个项目的整个实现过程非常不容易。项目刚开始的时候只有3个成员，而且都是刚加入的新人。而在Google，公认的说法是新人进入公司半年之内，无法很有效地进行开发。记得我带领团队，从3个新人，扩充到20来人的研发规模，不仅发展得很快，而且团队气氛也一直培养得很好，甚至后来还被评为公司最佳团队之一。至于具体的困难，我觉得最棘手的，并不是技术上的问题，而是观念上或者协调上的，比如说跟总部在理念上的冲突而造成的一些障碍。实际上，我们这个产品如果不是因为在一些“检查（Review）”上耗费了大量的时间，可能会早半年发布。

记得当初，我就是想把谷歌音乐做成一个与众不同的产品，而不是一个百度MP3的复制品，所以我一直在尽力推出一些与众不同的功能，比如个人推荐、音乐泡泡……在这其中，有很多都是探索性的功能，能让人眼前一亮。我觉得一个产品如果处处和其他产品雷同，就意味着失败。

记者：后来您为什么加入MediaV？作为CTO您如何规划公司的产品技术方向？

胡宁：当时，我一朋友告诉我，上海有一家公司正在招聘CTO，职位非常适合我。由于MediaV在上海，而我在北京，所以并不是特别想去。但在对MediaV进行了深入了解后，让我感到非常惊讶，尤其是它的CEO杨炯纬不仅对整个互联网广告行业拥有准确的洞察力，而且理念跟我非常一致——我们都相信技术创新是公司的核心价值。但这并不是所有人都相信的，尤其是在中国。

我目前的工作重点是把我们下一代的广告系统架构搭建起来。目前，我们公司已经跟复旦大学计算数学系联合创办了海量计算研究所。我觉得这在广告公司里面可能是绝无仅有的，因为我们公司更倾向于把自己定位成互联网技术公司。此外，我们还资助了一个关于国际数学家的奖项。

光听领导的话未必是好事

记者：您对技术人员有哪些建议？

胡宁：第一，要打好基础，这是以后能在技术上进一步提高的必要条件。第二，每次做事情的时候，都要思考一下为什么要这样做。我发现中国的工程师有一种倾向，就是比较听领导的话，这在某种程度上是好事，但如果是只听吩咐、光做不想，也是有问题的。如果你每次都能想一想为什么要这么做，不断地归纳总结经验，自己的水平就会很快提升。第三，多接触各个方面的信息也很重要，比如CSDN举办的各种技术交流会。我自己也经常去JavaEye阅读文章，感觉很受益。所以我提倡技术人员应该多积极主动地进行多方面的交流，而不是“闭门造车”。

兴趣是最好的工作动力

记者：作为一个女孩子，您对IT行业中的女孩子有哪些建议？

胡宁：我曾经与北大计算机系的女生们座谈，发现很多女生觉得自己不应该选择计算机专业。我觉得这与我们现阶段的社会大环境有关。许多人都觉得女生做技术，受到的压力会比男生要大。现在做IT技术的，女生大约只占总人数的20%以下，而男生要占到80%以上；但是我觉得只要对所学的专业感兴趣，只要喜欢自己所从事的工作，女生也一样可以做得很好。而且，我也经常在IT行业里碰到一些女生，她们都非常优秀，并不比男生差。（记者/张雪峰）



中国开发者将迎来黄金十年

——2010 软件开发 2.0 大会报道

■ 记者 / 付江

12月9日—10日，由全球最大中文IT社区CSDN主办，盛大网络、《程序员》杂志、JavaEye社区协办的年度收官技术盛宴软件开发2.0（SD2.0）大会在上海光大会展中心国际大酒店成功举办，来自微软、盛大、阿里巴巴、Facebook等众多领先企业的技术领袖汇聚一堂，共同剖析了当前IT技术发展大势。

开场致辞中，CSDN董事长蒋涛阐述了技术分享的重要价值，并通过对一系列具体数据的分析，解读了当前移动应用开发和开放平台火热背后的深层原因，他表示：在终端+技术变迁带来的大背景下，中国开发者将迎来黄金十年。

主会场Keynote上，包括阿里巴巴集团首席架构师/阿里云计算公司总裁王坚、盛大在线首席技术官朱敬、新浪微博事业部副总经理芦义、微软Visual Studio商业软件部总经理潘正磊、DR globalDirect分部副总裁Mark Iverson、360公司副总裁/CTO谭晓生、云快线科技有限公司CEO/世纪互联集团副总裁左洪、Autodesk中国研究院核心产品和运营总监云浪生、Frost & Sullivan中国区总裁王煜全、NHN中国开发中心技术总监崔海等在内的众多业界领袖发表了精彩演讲。

关键词：云计算、移动、开放平台

在主题演讲中，**阿里巴巴集团首席架构师王坚**认为，技术的发展和云计算模式的畅行，让开发人员创业的风险和投入大大降低，而阿里云的目标就是为广大技术人员提供一组规模化、低成本且具有高度弹性的服务。王坚介绍了阿里云的三层服务体系SaaS/PaaS/IaaS各自的特色，他表示公司将加速在云计算和移动领域的布局。据王坚透露，明年第一季度阿里云计算服务将正式对外服务，第二季度将发布与云计算对接的移动计算平台。他特别表示，CSDN会员将是第一批受邀体验的用户。

盛大在线首席技术官朱敬在演讲中认为，目前中国的创新环境仍然非常恶劣，盛大愿意战略性地将旗下的各种资源和服务，以API的方式开放出来，为开发人员提供全面整体开放和创富机会。

微软Visual Studio商业软件部总经理潘正磊分享了微软基于云服务，在产品开发、数据



2010软件开发2.0大会专家论坛现场

中心搭建、全球网络部署领域的实践经验。她以微软Visual Studio开发过程举例，每次在发布前，都会对产品进行高密度的集中测试，包括不同语言、环境下的测试，届时会对机器和资源需求非常大，而平时又会很低。在这个过程中，微软大量采用了云计算相关的技术。潘正磊还介绍了微软云计算策略在国内落地、合作伙伴以及云计算应用孵化中心的最新情况。

新浪微博事业部副总经理芦义介绍了新浪微博开发架构和开放API的情况，他表示，新浪官方目前做了一些基础功能和应用，更多的可能性都留给了广大的开发者去创造，包括图像、视频分享、生活类、电子商务类以及利用微博特性的各种社交游戏，开发者都有广大的空间。此外，按照开放计划，新浪微博平台将给开发者应用提供广告资助，并通过微博平台为应用增值提供支付渠道。此外，新浪还联合投资公司筹集了2亿元基金款项提供给自由的创业团队。

把握市场，开拓创富新模式

DR globalDirect分部副总裁Mark Iverson在演讲中介绍了全球软件销售市场的最新情况，并对SaaS在线销售和应用商店等新型模式的实效性进行了分享探讨。他表示传统的零售软件模式除了游戏类还在继续增长外，其他方面的销售额则在下降，而全球消费者的在线采购软件行为大概以每年13.8%的速率增长。他重点谈了混合型的软件销售模式，先卖软件，再通过网络服务提供支持，不仅可以增加客户的黏性，开发商通过服务提供更高的价值，同时也提供给合作伙伴盈利的机会，他们都通过数据在第一时间得到回报和反馈，再通过反馈来加强产品，不断增加客户持续支付的期望值。

第九城市副总裁兼无线互联网事业部总经理沈国定就手机游戏社交化趋势以及引进国内的OpenFeint功能进行了分享。他表示，OpenFeint作为一个基于手持移动设备的游戏社交平台，对于用户来说，能为手机游戏提供类似于Xbox LIVE一样的积分、成就、排行等额外功能，并让用户结成类似Facebook般的人际网络。对于开发人员，只需要在游戏中简单添加一些代码，就能获得多种在线交互功能，从而把单机游戏升级为一款社交游戏。而通过云计算技术，所有的网络服务、用户数据存储、服务器架构维护都由OpenFeint平台承担。

云快线CEO左洪在题为《网富论》的主题演讲中认为，对于技术人员，云计算提供了低成本创业、极速部署、低成本试错、Web Service API及自服务、自选操作系统等诸多便捷服务和优势。同时，他表示云快线将专注于为广大技术人员提供基于云计算的基础设施服务，而不会去涉猎应用领域。

新体验，新平台，新机遇

Facebook高级研究科学家蒋长浩在演讲中介绍了Facebook性能优化团队如何在2009年下半年的6个月时间里，将网站用户体验性能提高了一倍。此外，他还谈到了Facebook如何定义和监测用户体验性能，着重介绍了BigPipe技

术的设计与实现， 并深入分享了支撑BigPipe的Pagelet编程模型、BigPipe的多种运行模式以及Facebook在应用BigPipe实践中的经验。

此外，360公司副总裁/CTO谭晓生就《客户端的云端之路》以及云安全软件架构特性，Autodesk中国研究院核心产品及运营总监云浪生就《技术与设计创新》，Frost & Sullivan中国区总裁王煜全就《无线相关创业机会分享》，NHN中国开发中心技术总监崔海就《面向服务的计算平台》，Mochi中国总经理黄骁虎就《内容云——跨平台游戏社区》以及盛大创新院研究员范路就《开放的电子书对开发者的新机会》均发表了精彩的主题演讲。

八大分会场，聚焦技术、产品与实战

承接Keynote主题，第二天共安排了八场内容丰富的分论坛，除Web开发、工具与语言、架构与实践、软件工程等已成业界品牌的经典主题之外，今年还特别新增产品与设计、云计算、社会化计算、开放平台、电子商务等热点话题，成为大会举办以来覆盖领域最广的一次。

分论坛演讲嘉宾阵容强大，IBM中国新技术研发中心总经理毛新生、英特尔亚太研发中心高级软件工程师张银奎、阿里巴巴国际交易技术资深总监龚义涛、全国海关信息中心架构师王翔、上海科泰华捷公司董事长陈榕、创新工场首席布道师/软件架构师蔡学镛、Parasoft China技术总监伍宏益、UMLChina首席专家潘加宇、人间网共同创始人/CTO曹晓钢、Magicome公司创始人西乔等各领域深受开发人员爱戴的技术专家云集，给参会者带来一场丰盛的技术大餐。



CSDN董事长蒋涛指出中国开发者将迎来黄金十年

在“云计算”分会场，来自于英特尔亚太研发中心的高级软件工程师张银奎用诙谐直白的语言，深入技术底层，重点介绍云计算的技术优势，以及支撑云计算的三大主要技术（互联网、虚拟化和网格计算）中的虚拟化技术。



会场嘉宾与参会者亲切交流

在“软件工程”分论坛，上海漫索计算机科技有限公司创始人兼总经理林锐发表了主题为《软件企业常见问题和系统性解决方法》的演讲。他解读了软件企业经常遇到的八大问题：包括软件挣钱难、组织结构和人力资源、跨部门协作、流程改进、需求沟通、设计和开发、软件项目管理以及管理工具问题。同时，林锐还分享了他在实战过程中积累的以上问题的解决之道，并与现场技术管理者展开了精彩的互动。

此外，其他分会场嘉宾也就擅长的领域发表了激情洋溢的演讲，获得了与会者的纷纷好评，现场QA环节更是精彩亮点不断。

写在会后：技术盛宴之后，人才战刚刚开始……

苹果的成功向业界证明：得开发者得天下，2010年软件开发2.0大会，再次验证了“人才是核心竞争力”的硬道理。

正如CSDN董事长蒋涛在演讲中所言，随着智能化移动终端的普及和百家齐放的开放平台战略的发展，厂商对于开发者的争夺已扩展到正在蓬勃发展的移动互联网领域，甚至很多过去被认为只做“硬件”和“服务”而与开发者相差甚远的企业，不但大力布局开发者战略，甚至直接加入到争夺开发人员的行列。而如何为开发者创造更具吸引力的平台、盈利方案和策略，将成为众多企业未来必须考虑的核心战略之一。

一场对开发者的争夺大战已经拉开，你做好准备了吗？

软件企业常见问题与系统性解决方法

■ 文 / 林锐

本文的目的是让大家了解软件行业普遍存在的问题，而不要等到出了问题之后，再头痛医头、脚痛医脚。软件行业需要系统性的解决方法，避免常见问题一而再、再而三地发生。更详细的阐述可以参考我的著作《如何管理软件企业》第2版。

软件企业面临的常见问题

做软件挣钱难的问题

千万不要以为做软件是一本万利的事。中国目前绝大多数软件企业过着苦日子。软件业主要有三类盈利模式。

一、承接合同项目，为甲方开发软件系统

如果软件公司从自由市场上承接合同项目，将遭遇如下困难。

- 能承接项目的软件公司乃至个体户实在太多，客户会把合同项目的金额砍得很低，单个项目的利润空间本身就小。
- 项目需求和验收受制于客户。开发过程中，客户会不断变更需求，导致开发方不断修改软件，项目验收被不断地延后，开发方成本（包括机会成本）越来越高。
- 缺乏规模复制效益。

二、开发并销售通用软件产品

目前凡是面向个人的通用软件产品，由于盗版原因，几乎无法靠卖软件来挣钱，只能改变盈利模式。只有企业级软件产品，因其推广使用错综复杂而不容易被大量盗版，可以走“通用软件产品盈利模式”。但想做好通用的企业级软件产品，难度非常高，因为企业级软件的复杂度远远高于面向个人的软件。开发方必须把自己打造成为“企业级应用的领导者”，否则潜在客户不信任你的产品，那么产品就无法通用。

三、运营模式

不承接合同项目，也不卖软件产品，而是通过互联网或移动通信网做运营，靠用户频繁使用软件来盈利。主要存在以下问题。

- 互联网公司的业务太容易被模仿，同质化竞争严重。
- 互联网公司的另一大缺点是浮躁，过分追求快而导致根基不稳，据我所知，国内大多数互联网公司的软件研发管理，要比传统软件公司混乱得多。

任何一个软件产品，诞生它就要花上一年半载，推向市场后还要花多年心血才能把它真正做好。软件企业要想多挣点钱，就要设法“开源节流”。“开源”主要靠优化盈利模式，使得现有的技术和资源产生更高的效益。“节流”主要靠改进管理，使企业的所有经营环节更加合理，减少不必要的成本。

组织结构和人力资源问题

- 组织结构臃肿，工作效率低下。岗位和职责不清晰，责、权、利不明。
- 项目矩阵关系比较复杂，项目成员不知道是听“职能经理”还是“项目经理”指挥。
- 不能知人善用，无法发挥团队中每个人的价值，老抱怨员工素质低。
- 重要岗位用错人，例如把技术水平很高但是情商低的人提拔到领导职位，不仅荒废了这个技术高手，而且带乱了队伍。
- 优秀人才难招到，也难以留住。难以准确地评价研发人员的业绩，缺乏有效的激励措施。

跨部门协作问题

- 跨部门人员相互不熟悉对方的工作流程和规范，沟通费劲。
- 上游传达给下游的需求不够清楚，各方理解有偏差，上游不断变更需求，导致下游不断修改，频繁浪费。
- 由于各部门的目标和利益不同，导致跨部门合作产生矛盾。例如营销部门为了使客户满意、增大销售额，答应了客户太多的需求，致使研发部门耗尽有限资源，疲于奔命。

企业不仅要让每个部门制定内部工作流程，还要把各部门的流程整合起来，形成整个企业的集成化流程。下游一定要有制约上游的机制。

流程改进问题

大多数软件公司成立最初几年，几乎没有流程，基本靠人治。等公司规模稍大后，就基于CMMI搞流程改进，不知不觉把软件流程搞得很复杂，貌似完善却难以推行，走向了另一个极端。软件公司既希望有完善的流程来指导重大项目的研发，规避风险，又希望有快捷的流程适用于“短平快”的项目。软件公司需结合CMMI理论和敏捷开发思想来优化流程。

需求问题

客户方存在的问题是，客户说不清楚需求，双方误解需求，客户经常变更需求。

开发方存在的问题有以下方面。

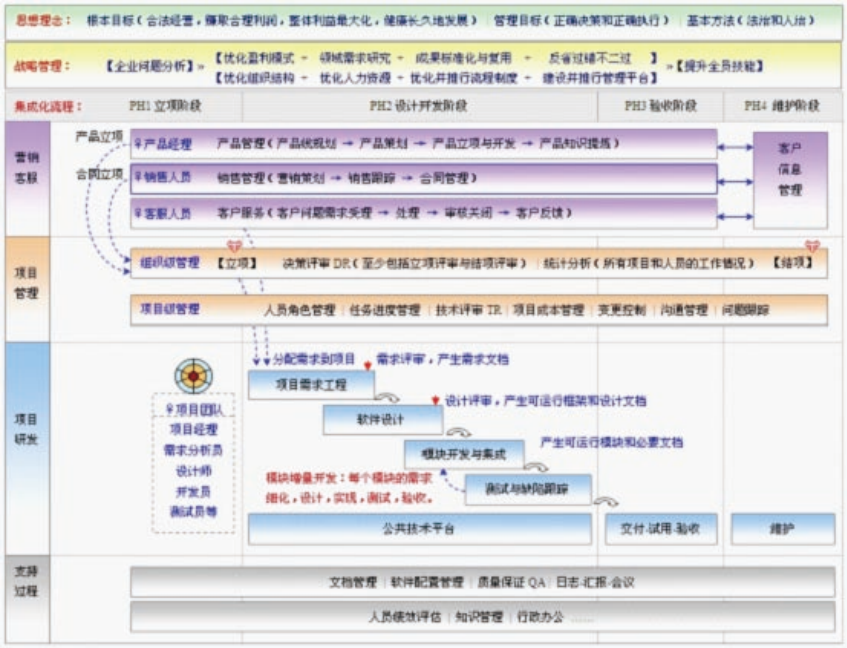


图1　集成化管理模型

● 没有熟练掌握需求工程的各项技能，如需求调研、需求分析、需求定义、需求评审、需求变更控制等。

● 公司没有在战略高度上重视“领域需求研究”，仅仅把需求分析当做项目中的一个环节看待。每个项目都从零开始做需求分析，被动地等待客户提出需求、变更需求。而没有主动研究领域需求，引导客户消费。

设计问题

● 同一系列的产品、甚至一个产品之内，采用了多种编程语言和多种差异较大的技术。受制于原有的落后技术，先进技术难以无缝引入。

● 在很多项目中重复地开发相似的功能，互不通用，浪费生产力。没有提炼标准件，没有建设公共技术平台，就无法取得软件复用带来的高效率。



图2　软件企业战略管理模型

● 软件用户界面设计是大多数软件公司的弱项，人们不知道怎样才能设计出易用、美观的用户界面。

● 软件设计应当“细到什么程度”很难把握。

软件项目管理问题

软件项目管理的独特之处是，它是建立在软件工程基础之上的专业化项目管理。一个没有软件工程经验的人，即使通读PMBOK也管不好软件项目。

软件项目管理的范围主要包括：立项管理、结项管理、任务进度管理、项目评审、变更控制、软件质量管理、软件配置管理和文档管理等。

管理工具问题

大部分软件公司没有对管理工具进行规划和统一部署。各部门、各项目采用自己熟悉的管理工具，分别用于管理客户问题需求、项目信息、任务进度、代码库、文档库、测试与缺陷跟踪等，管理工具之间各不兼容，形成信息孤岛。有些工具很老旧、不好用，但是由于有数据存在，既不能淘汰，又不能更新。结果这么多杂乱的工具非但成不了财富，反倒成了包袱，十分头痛。

企业需建设与流程配套的集成化管理平台，监控所有项目和人员的工作情况，不断积累知识财富，提供更高级的统计分析和决策依据。

软件企业集成化管理模型

我经过十多年的方法论研究与实践，提出了系统性的软件企业管理方法：集成化管理模型，如图1所示。

思想理念层。目的是让企业全员对“是非对错、轻重缓急”有共同的认识。有了这个基础，企业才可以推行具体的管理措施。

企业战略管理层。目的是使企业持续进步、长治久安，是企业领导层最主要的工作。

集成化流程和工具层。把软件企业的主要执行过程——“营销客服过程、项目管理过程、研发过程、支持过程”集成在一起，使企业流程通畅无阻，各部门人员有条不紊地开展工作。

软件企业战略管理：持续进步十项措施

此外我总结了使软件企业持续进步的十项措施，绘制成为“软件企业战略管理模型”，如图2所示。

（本文整理自作者在2010软件开发2.0技术大会上的演讲。）

作者简介　林锐



上海漫索计算机科技有限公司创始人兼总经理，致力于创作适合国内企业需求的研发管理方法论和软件产品。累计出版《高质量程序设计指南——C++/C语言》、《软件工程与项目管理解析》等8部图书。

■ 责任编辑：常政（changzheng@csdn.net）

架构重构趋势谈

■ 文 / 温昱

近几年，我专职从事为软件企业提供咨询和培训服务的工作，所以得以近距离观察众多一线软件研发团队的喜怒哀乐。其中，重构已经成为目前软件研发的关键环节，本文谈一谈我所感受到的架构重构发展趋势。

难点 · 痛点 · 未来热点

“架构师就是设计架构的人”，这种理解太简单化了，没有反映出不同背景的软件企业、不同发展阶段的软件企业所重点关注的“主战场”的不同。我用一张图来刻画架构师的几个“主战场”（如图1所示），以辅助我们更准确地“定位”架构重构在架构师工作中的位置。

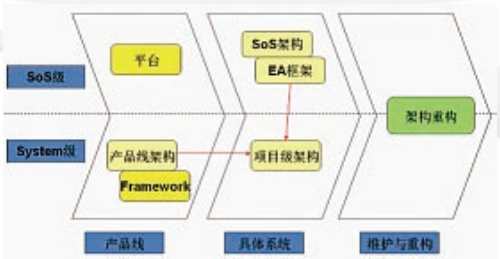


图1　架构师的主战场

随着不同产品的推出、不同版本的发布，需要维护的遗留代码越来越多，重构也就在所难免。关于架构重构能力之于软件企业的意义，可用八个字概括：难点、痛点、未来热点。

难点。不少软企都有架构重构的意愿，但经常是一拖再拖不敢实施。进行了架构重构之后，也有企业发现没效果——架构质量没有得到改善——这相当于架构重构失败了。这是因为，架构“重构”是难点，它比架构“设计”更难。

痛点。困难就不干呗？可是不行。加个Feature很“难”，改个Bug很“绕”，软件工程师们被加班搞得很“惨”……软企管理层也倍感压力，因为维护成本日益呈现攀升趋势，“加快问题单响应速度”目标的达成也越来越遥远。“如何把架构重构好”，成了大家共同的痛。

未来热点。既然是不好对付的“难点”，又是影响软企切实利益的“痛点”，架构重构领域就必然是“未来热点”了。

种种迹象

下面简要梳理一下“重构能力越来越关键”的种种迹象。

对个人而言，重构能力影响着研发人员的工作业绩、职业发展，是不折不扣的“核心竞争力”。因为当前业界越来越重视对遗留代码、第三方代码、开源代码的利用，掌握重构能力的研发人员能在竞争中脱颖而出。相反，不能自由掌控代码的程序员，加班不少、业绩不高；对现有不满意的架构“力不从心”的架构师，工作也处处被动，高薪但不开心。如图2所示。

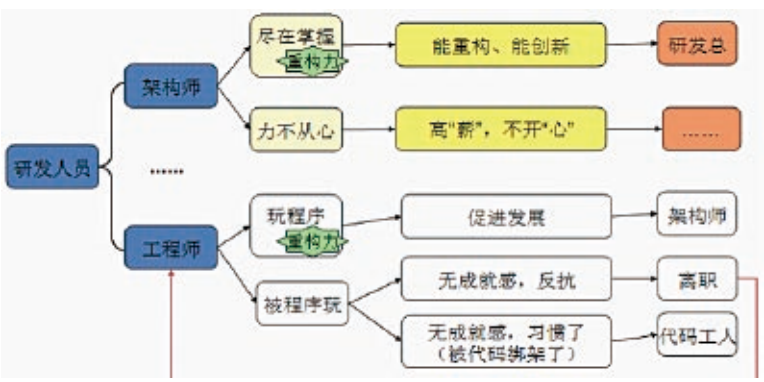


图2　重构能力与个人发展

而软件企业，早已有意无意地开始了“重构人才争夺战”，不断在招聘广告中打出如下要求：

- 对已有系统进行重构和优化；
- 对组件的重用、重构有丰富的经验；
- 能够熟练运用各种重构方法；
- 熟悉Linux系统重构、Bootloader移植；
- 察觉实现问题，提出改进（重构）方案；
- 对框架本身的体系有较为深厚的理解和应用经验，对框架本身有过开发或重构者可优先考虑

……

同时，大型软件企业也已越来越关心开发骨干重构能力的培养，我从2006年专职从事咨询和培训以来的服务经历已证明这一点。其中，2010年我帮助大型软件企业提升重构能力的服务更有47个工作日之多（9周）。

未来3–5年的具体趋势

软企面临的实际问题以及相应的实践探索，



图3　未来3~5年的具体趋势

是推动未来发展的根本原因。如图3所示，是我对未来3–5年具体趋势的判断。

趋势1：认识趋于专业——3种重构的细分

当前，对不同层面重构的明确认识还不普及，有很多错误观点在流行。例如，诸如“架构重构和《重构》书里说的代码重构差不多”等观点，是不切实际的。更专业的认识，是将重构分为代码重构（Refactoring）、模块重构（Big Refactoring）、架构重构（Architectural Refactoring）三大类（如图4所示）。

“不同重构技能混为一谈”造成的危害，在软件企业中相当常见。其中，软件研发人员们“小重构不屑做，大重构不敢做”的表现，是最令研发管理者头痛的。你们公司，有吗？

更进一步，“重构的3个问题，2.5个亟待发展”是我们的基本判断，即：代码重构发展不错但还将继续延展，模块重构和架构重构发展则严重不足。具体观点分别在下面的趋势2、趋势3、趋势4中简述。

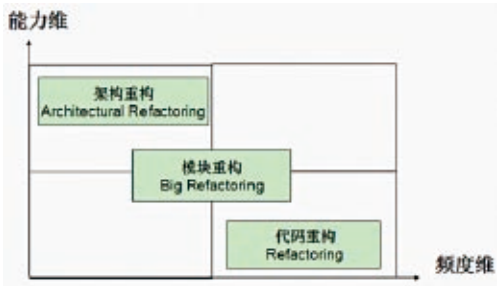


图4　认识趋于专业，3种重构的细分

趋势2：代码重构——往特定领域纵深发展

当前的代码重构成例的总结，总的来说多在OO方面。但通过分析一些大型软件企业的领域特点和代码现状，我们发现大量遗留代码甚至是十年之前的，既有C语言编写的结构化代码、又有C++编写的结构化设计思想的代码（只有成员变量或成员函数的类大量存在）。因此，未来必然有更多、更具针对性的重构成例在软企的维护一线出现。

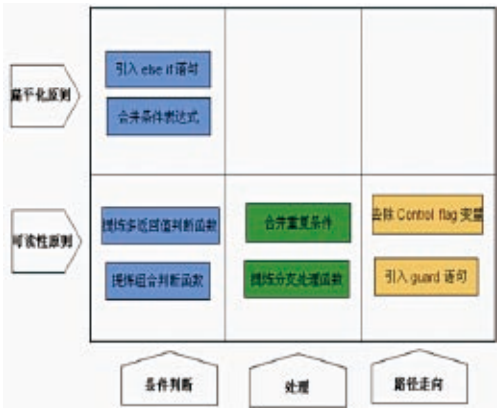


图5　复杂条件的重构成例地图

例如，在和大量软企及其结构化代码的接触过程中，我们总结和“部分原创”了下列“复杂条件重构成例地图”（如图5所示），在重构咨询中有良好的应用效果。

趋势3：模块重构——软企能力建设“热点”

趋势4：架构重构——切实有效的架构重构方法

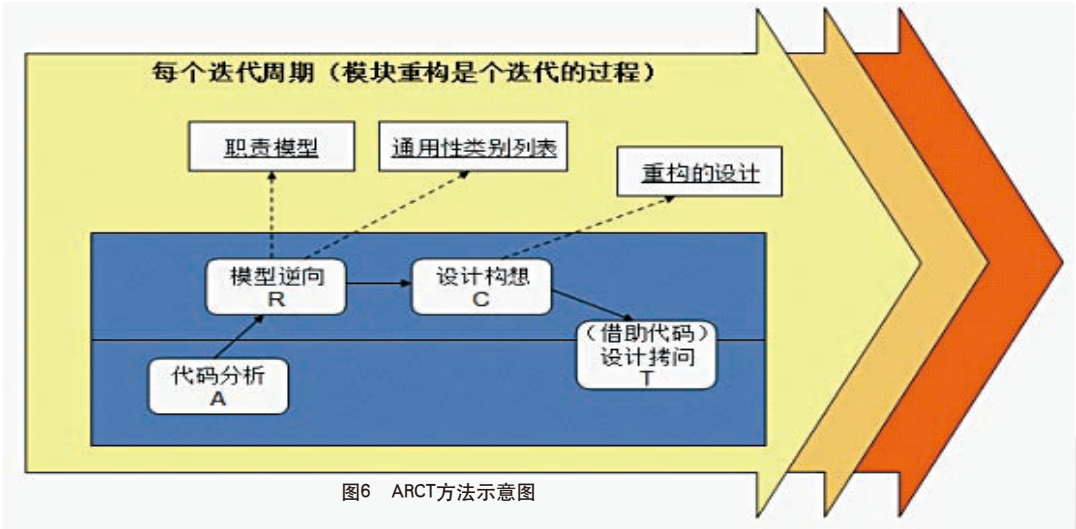


图6　ARCT方法示意图

过去，模块重构和架构重构发展严重不足，一线实践往往找不到可参照的方法，多靠摸索来实践。例如，有架构重构方法提出，首先进行的是“重构计划制定阶段”，这种方法在软企中是没法用的。（不分析代码、明确问题，怎么可能订计划？）

未来3–5年，这一块将有突破，从领先的实践中孕育出切合实际的模块重构方法和架构重构方法。

趋势5：五年内——出现专职“设计重构师”职位

《程序员》杂志的各位读者，可以做个见证。

软企提高重构能力的几点建议

这里我给两点建议。

1. 架构重构，必须重视代码。

2. 关键模块的重构，可以基于ARCT方法论，迭代式开展。

虽然架构比较抽象，但“借助架构文档等进行架构重构”的做法是不务实的。现实是，代码中有随处可见的“贴膏药”式的特殊情况处理，有时更是“补丁摞补丁”——这些都是文档中没有覆盖的。这些“膏药”和“补丁”恰恰是现存系统正确处理业务的“宝”。**所以，我们的架构重构“不得不”依靠代码。**

另一方面，又不能陷入代码泥潭。我们的应对理念是“基于7种接口风格的模块任免”，篇幅所限，本文就不展开了。

有经验的架构重构实践者会发现，架构重构非常可观的工作量，是用于进行核心模块的重构。根据我们的经验，模块重构最终经常会落实成几十个代码重构（Refactoring），但这绝不意味着模块重构就是代码重构的简单累加——设计重构的方向在哪里？模块重构设计师必须有能力解决这个问题。

ARCT方法是我们逐步实践成型的一种模块重构方法论（如图6所示），已多次用于大型软企的关键岗位能力建设咨询，并取得成功。

代码分析（A）是基础。从代码入手。

模型逆向（R）是纽带。从代码级思维跳到设计级思维，抽象出当前的设计模型。

设计构想（C）是关键。例如，为了获得更好的可扩展性等，新设计要重构成什么样呢？……

设计拷问（T）是保证。回归代码层面进行设计可行性、合理性的主动质疑。可以工作吗？可以正确处理那些散落在代码各处的特殊处理吗？适应了各种场景带来的可扩展性、可维护性的压力了吗？模块重构时，必须适时思考这些问题。

作者简介　温昱



软件架构专家，资深咨询顾问，实战型架构培训专家，ADMEMS架构实践体系创立者。拥有十年系统规划、架构设计和研发管理经验，出版过《软件架构设计》、《一线架构师实践指南》等专著。

责任编辑：常政（changzheng@csdn.net）


TUP
Technology
User Experience
Product
技术
体验
产品

TUP：成功产品三要素

■ 策划 / 本刊编辑部

今天，产品已经成为市场竞争的核心，产品的成败关乎研发人员、团队乃至整个企业的兴衰。要想做出一款成功的产品，必须具备三个基本要素：技术（Technology）、用户体验（User Experience）、产品运营管理（Product），我们将之简称为TUP。

技术显然是成功产品的必备条件。开源技术、云计算基础设施和开放平台的快速发展已经大大降低了产品实现和传播的门槛，让好的创意可以在很短时间获得数以百万计的用户。但是，如果你无法解决性能、稳定性、安全和容量等前端和后端的技术问题，会对用户体验产生灾难性的影响，最终在激烈竞争中失败。

用户体验对于产品成功的重要性更是不言而喻，近年来取得巨大成功的产品几乎无不具有令人惊艳的用户体验。从某种意义上来说，用户体验绝非UX设计与工程人员的专项，它其实无所不在，除了体现在与最终用户使用相关的便利性和美感、舒适感与情感等层面之外，也渗透在技术研发协作中的方方面面，你取的变量名是否好记不混淆、你设计的API是否易用、你写的文档是不是好懂、你讲的话是否能让对方真正理解……

“优质产品是运营出来的。”在2.0渗透到社会各个领域的当今时代，用户其实是在和我们研发人员共同驱动和塑造产品。但是，如何对需求和最初的创意进行分析、规划并指导研发与设计，如何对产品进行营销和推广，如何通过现在各种平台和渠道获取更多忠实而且有建设性的用户，如何营造创造性氛围形成活跃的社区，如何从用户中汲取宝贵的集体智慧，却不是简单的事情。产品的运营与管理是一门艺术，它为技术与用户体验指引方向，实现完美的结合。

2010年，《程序员》杂志联合CSDN和JavaEye推出了TUP为总品牌的系列线下活动，主要针对IT产品研发相关的技术、设计、运营、运维、管理等专业人士，涵盖IT产品研发背后的成败经验。活动分别在北京、上海和天津举办了多次，受到社区的广泛欢迎。12月举行的2010软件开发2.0大会也以TUP为主要的关键词。同时，杂志中除了原有的研发技术与实践之外，也开辟了TUP专栏，并增加了用户体验与产品方面的内容。2011年我们将在保持技术特色的同时，继续加大这些领域的力度。

变革中的技术

——2010-2011回顾与展望

■ 记者 / 董世晓



毛新生

IBM中国新技术研发中心总经理、IBM中国开发中心CTO。先后从事信息检索、Web Service、SOA、Web 2.0、云计算和行业解决方案等领域的工作。



杨卫华

新浪产品事业部技术经理，专注于开发高并发的分布式应用。对互联网后端技术、分布式、网络编程、XMPP即时通讯等领域感兴趣。



王宏

大众点评网技术部系统架构师。对互联网的前端技术、系统平台架构等技术方向有丰富经验，喜欢Coding和Troubleshooting。



李锟

阿里巴巴集团B2B子公司平台技术部技术专家。14年工作经验，8年Web应用、3年企业应用、3年电信协议的实践经历。



范凯

CSDN产品总监，负责CSDN网站和JavaEye网站的产品设计、研发和网站运营。JavaEye网站创始人，资深软件工程师，12年从业经验。

您认为2010年重要的技术热点有哪些？为什么？

毛新生：我认为，在企业计算领域，**云计算**、**物联网**和**业务分析**是2010年主导的技术热点。云计算已经成为了现实，不再停留在概念上。从基于云计算平台的软件开发云和测试云，到云管理和应用健康管理，再到统一通信以及协作平台到云计算的迁移，无不是云计算推动的技术创新与业务变革。物联网的远景目标是组成一个超大的物品智能网络，由此推动城市、医疗、建筑等的智能化。作为物联网应用的智慧城市将数字技术应用到物理系统中去，并利用所有产生的数据提高生活的空间、效率与质量。业务分析对特定业务问题涉及的历史的、当前的和预测的数据进行深度的挖掘与检视。对客观数据的分析洞察能指导我们的行动以获取最佳效益。这些基于事实数据的洞见与行动指导，将为系统和业务人员提供普适的决策支持。

杨卫华：NoSQL分布式存储依然是一个热点，尽管尚未有非常强势并成熟的开源解决方案。以Cassandra为首的Dynamo类型分布式存储产品发展遇到一些稳定性问题，业界对其研究有放缓的迹象，其中的标志事件就是Twitter暂停在主业务存储使用Cassandra。在11月，Google的Jeff Dean在Standford的一次演讲中又提到分布式系统中增加一个中心状态节点的好处，业界或许会重新反思Dynamo这种去中心化思想在工程实践中是否比中心化更容易出现稳定性问题。尽管NoSQL分布式存储得到了飞速发展，但是主流海量数据存储仍然严重依赖MySQL等传统数据库方案来解决。

性能是Web架构永恒的主题，强调性能为

王的Velocity大会也在年底来到了中国。其中反响比较大的是Facebook的前端架构优化主题。Facebook使用BigPipe技术将网站请求速度提高了2倍，它的设计思想在很多方面值得借鉴，包括在服务端的设计。目前业界大部分Web系统都是串行化设计方法，如果借鉴BigPipe思想改成多个Pipeline并发执行，可以实现在相同体系架构、相同软硬件环境下访问速度的很大提升。

其他一些热点如Cache中心化设计我在博客及新浪微博上已经有较多介绍，有兴趣读者可以进一步去了解。

王宏：在我眼中，有三大技术热点。

第一，云计算。云计算将所有的计算资源集中起来，“云”通常使用一些大型服务器集群，例如计算类、存储类、网络类等，通过软件实现自动管理、按需分配，而且这种资源在使用者看来几乎是无限的，只要需要都能通过简单的资源配置实现应用。这使得使用者无需为复杂的服务器配置及其相关优化而烦恼，能够更加专注于自己的业务应用。这种方式使单个服务资源得到最大化利用，节省资源。

第二，虚拟化。随着服务器的运算能力越来越强，资源越来越丰富，虚拟化技术也越来越得到大家的重视。通过虚拟化技术，可以实现在单一的计算机（服务器）上安装多个操作系统或者多重任务的处理，这样既能达到高速处理业务的目的，同时又节省了企业的IT成本。

第三，HTML5。如果说前几年是HTML5的诞生之际，那么2010年就是HTML5的成人之年。今年HTML5迅速普及，并且在移动设备上得到快速的应用。HTML5使得原有的网页变得越来越“丰富”，原本通过Flash、Media Player、QuickTime等专用插件才能使用的视频、动画变得易用，而且有着更好的兼容性。可以说，HTML5让开发人员从不同的插件、不同的技术以及不同的设备上得以解脱。

范凯：依我之见，2010年的热点技术无非是以下三个方面。

第一，移动应用的迅速普及。2010年是智能手机和移动应用迅速普及的一年，年初苹果的iPad发布，掀起了手持式智能移动设备普及的浪潮，Android也开始四处开花，不仅仅在智能手机领域，还在很多移动领域（如车载GPS）迅速普及。总的来说，2010年是移动时代引爆的一年。

第二，HTML5技术的普及。得益于苹果iPad的推动以及大多数厂商的支持，HTML5技术在移动领域方兴未艾。

第三，NoSQL领域仍然在迅速发展。尽管Cassandra的应用遇到一些挫折，但是像MongoDB和Redis这一年有了惊人的发展，在越来越多的网站包括CSDN中得到应用。

2010年值得分享的新工具有哪些？

毛新生：云计算方面的工具有IBM Cloudburst、BLADE VMready、HSLT、LotusLive等。物联网方面的工具有Smart Food、Smart Water、Smart Traffic、Smart Healthcare等。业务分析方面的工具有Cognos10、Openpages、SPSS DM、Guardium等。

王宏：iPad，不能简单地算工具了，而是因为它衍生的很多软件，例如pWiki。

使用iPad能随时随地上网检查各种服务的状态，并且可以简单地处理一些问题。

您在实际工作中面临的最大挑战是什么？是怎样解决的？

毛新生：技术服务于业务，如何通过整合提升成熟的技术系统，服务于客户日益增长的业务需求，是技术研发团队面临的最大挑战。通过与Marketing、Sales、Consulting部门的紧密合作，走出实验室，与客户展开广泛深入的多层次的合作，释放实验室的力量，遵循先由外而内、后由内而外的设计方针，敏捷化研发流程，是我们认为战胜挑战的根本方式。

王宏：最近帮助手机团队构建手机通信的API服务，其中遇到很多问题，例如各种网络环境、硬件环境下的通信速度问题、移动设备的辅助定位问题、GPS定位后获取的经纬度与地图上的经纬度偏移的问题等。具体的解决办法可以参考我在2010软件开发2.0大会上的演讲。

杨卫华：今年我工作中遇到的最大的两个问题是，系统的故障容错处理以及海量数据设计。



故障处理遵循的一个原则是Design for Failure。一个大系统要达到每个环节都有良好的容错能力，实现起来会有很多困难。大型的系统都不是一个模块组成，Jeff Dean提到在Google.com执行一次搜索会调用到100多个内部服务，新浪微博同样每一个环节牵涉到内部多个调用。因此如果没有良好的自动容错处理能力，很容易就会导致用户访问系统失败。可以说，对这一点的执行力决定了架构成败。

海量数据设计也面临挑战。分布式NoSQL存储目前还没有达到我们期望的稳定成熟产品。而使用MySQL拆分来存储海量数据存在很多已知问题，前期需要谨慎的设计，否则达到瓶颈后无法扩展将会给系统带来无法修补的灾难。上线后运维也需要很多资源来维护，包括冷热数据迁移、容量规划等。运行一定时期的优化又需要根据业务访问特点调整结构及切分方式。MySQL下的海量数据调整和重构做不到非常敏捷，调整的时间周期会很长，很多技术成本都消耗在上面。因此希望将来能够设计更多自动化系统及更好的产品来满足业务海量数据存储需求。

范凯：我所遇到的问题是，业务系统多而且杂，有各个时代、用各种语言开发的东西，而现在需要逐渐统一化，建立统一平台。

我的解决之道就是REST，各个系统模块化、松耦合，使用REST架构来编写，通过HTTP互调用实现系统的整合。

2011年有哪些技术趋势？

毛新生：2010年消费电子产业取得了飞速发展，智能设备和物联网的发展使得软件被普及到各种不同的设备中，影响到了每个人的日常生活，可以预见，这种趋势在2011年将进一步发展。平板电脑、智能手机、网络电视等为我们带来了更智能化生活的同时，也对信息系统提出了挑战：对于多种设备的支持，新的安全性挑战，更快捷地响应变化的市场，更有效的协作平台，信息全生命周期管理，高效的开发工具变得比任何时候更重要。越来越多系统引入商业智能与业务分析是2011年另一个可以预见的趋势，对来自内部、外部业务系统的数据进行分析、整理和预测可以帮助发掘业务信息中潜在的价值，从而指导业务绩效监控，提供更普适的业务决策支持。海量数据分析、实时数据分析、社会网络分析、风险与合规分析的开发应用也刻不容缓。

王宏：我看好近场通信（Near Field Communication，NFC），它是由免接触式射频识别演变而来。NFC不需要复杂的设置程序就能进行数据通信，而由于其耗电量低、一次只和一台机器连接，拥有较高的保密性与安全性，在不同的场合、不同的领域和其他通信方式起到相互补充的作用。其易用简便的方式让用户更能接受并使用，随着移动互联网的兴起，这种快速、便捷的通信方式越来越得到大家的重视。Google新出的手机已经支持NFC芯片，并有了相关的应用。NFC带来的不仅仅是单一的技术，更开拓了大家的思路，利用图形（二维码等）、声音等多种方式也能实现近场通信，相信在LBS、移动支付等方面会有更大的发展。

李锟：2011年我看好下面的技术趋势。

移动应用，尤其是LBS应用的普及，将会为我们的生活带来很大便利。

随着IE9正式推出，HTML5的热度逐渐高涨，将基于Web标准的RIA技术推向一个新的阶段。

REST开发会成为一个技术热点。不仅仅局限于Web服务，REST在企业应用中的适用场景，也会越来越多地被发掘出来。

Martin Fowler的那本讲DSL的书出版后，DSL也会成为一个技术热点。

用户为中心的物联网系统设计

■ 文 / 王翔



作者简介：

王翔，软件架构师，主要研究方向为XML、.NET、领域设计和PKI应用。工作之余喜爱旅游、写作和烹饪。

本文中的“用户”不仅是“人”这个个体，还包括物理空间中的“人”及其所驾驭和使用各类的设备、工具等。例如对一个SNS应用，其用户为中心设计往往指用户在网页中的各种体验；但在物联网应用中，一个场站闸口管理系统它的用户往往指开着车的司机或者是通过OCR读取车牌的软件或人（为便于表述，本文多处用“@H”标识“用户”或“人”，用“@T”标识“物”）。

现阶段，作为一个新的IT商业领域（尽管有很多同行认为它不新鲜、新瓶装旧酒，但作为一个商业领域，最起码作为一个概念姑且算新），物联网产品和应用的开发机构还将很多注意力集中在具体技术标准和商业模式上，但从趋势看后续跨“物”甚至跨Internet of Thing/Web/Mobile的交互标准以及用户行为标准可能更具商业价值。本文正是侧重于在“用户”层面为物联网应用设计者提供参考。

物联网应用运行机制分析

典型的物联网应用的执行过程如下。

- 对物体属性进行标识：现在的应用中，我们往往一眼就知道i是个变量，但物理世界没有这么简单。通过何种属性标识车辆、人、货物方法各异，比如超市可以用条码对货物标识，人可以通过二代身份证或声音、掌纹，车辆可以通过车牌。这一步解决的是“有的放矢”的问题。
- 识别物体属性：这步是物联网应用运行时的起点，它根据属性识别@T和@H，获取其指令。
- 将信息转换并通过网络传输。
- 由信息系统完成信息处理。
- 反馈：通过各种可能的方式给予@T或@H以反馈。

整体看，过程并不复杂，但关键在于“反馈”是否合适，尤其对具有主观感觉的@H而言，反馈方式是否合适就成为影响物联网应用生存的关键，就好像邮件客户系统很多，但真正具有大量用户并且获得利润的却寥寥无几。

用户为中心@thing

物联网具有泛连特性，但对于置身其中的用户，则存在如下问题。

- 多入口。
- 安全机制烦琐。
- 跨网络、跨应用协同困难。
- 访问手段单一。
- 信息因系统的隔阂被尘封。
- 难于集中管理和运维支持。
- 开发团队需要反复制造车轮和铆钉。

如果说在现有的应用时代，每个知识工作者平均面临20个左右的信息系统，在物联网时代的早期（2010-2020年），你将在潜移默化间主动、被动地面临600~2000个物联网系统。如果让用户“顶着”这些问题使用信息系统，结果是用户为系统所累，“刘姥姥进了大观园”。究其根本在于现阶段多数物联网信息公司的设计是以机器为中心（Machine-centered Design，MCD）。下面我们调整设计思路，将用户为中心设计（User-Centered Design，UCD）用于物联网系统。

普渡大学可用性测试检查表将UCD标准分为如图1所示的八类。

在物联网时代，他们在现有要求外还可以演绎出其他内容，下文我们就这些演绎的特色



图1　普渡大学可用性测试检查表

内容一一分析。

兼容性

很难用一个标准评价设备是“好”还是“不好”，因为所谓兼容性是区分上下文的。以一个办公室智能调节系统为例，如果你的用户是普通的商务人士，他们可能觉得将系统与现有的空调、电视遥控器结合不错，百叶窗采光除了可以自动调节外，允许用户通过遥控器调节。但对于苹果的粉丝，您可能更喜欢触屏系统，光线调节通过多点触摸直接在触屏完成即可。

物联网时代兼容性设计不仅要考虑@T部分，对于@H交互的部分不是说越Cool、越绚丽越好，要兼容用户习惯，而不是让用户兼容系统。

此外，根据项目经验，大部分情况下物联网应用中鼠标和标准键盘是无用的，在成本允许的情况下，与其让用户通过鼠标、键盘操作系统，不如增加简单的前端传感系统或机械控制系统。试想，当你驾车进入高速公路时，是给你套键盘、鼠标，让你输入车牌号 + ETC卡号方便，还是让你直接刷卡方便呢？

一致性

红绿灯的含义相信在绝大多数情况下是没有歧义的，但物联网目标对象无论是@T还是@H，它们的变化往往比下面这段代码执行过程复杂得多。

```
s=false;
if(s) pass();
```

因此，在物联网环境中，黄灯的含义如何界定就非常关键了。通过日常经验可以知道，黄灯可以通行，但你系统的所有功能中是否“黄灯时刻”都代表通行呢？即便你的系统这样，你集成的其他物联网、互联网应用是否也是这个标准呢？



图2　红绿灯

这时候，一致性很关键，须知一个电子商务应用的错误可能导致“人活着呢，钱没了”，但一个物联网应用出现一致性问题则可能导致“人死了，钱没花了”。

灵活性

再看一段再熟悉不过的代码：

```
temp = A[i];
A[i] = A[j];
A[j] = temp;
```

这段代码是排序中常用的交换，但如果请您在上班高峰期在国贸桥和前面的车交换是否一样容易？很难了吧。如果前面car[i] 的油突然等于0了，情况就更糟糕了。究其原因在于物联网世界中物体都是占有物理空间的。电子商务应用中的购物车，add(productID, quantity)只是一个方法，但现实中让自动化立体仓库中的机器人把库房两侧的冰箱、洗衣机放在一个大购物车（集装箱）需要涉及一组传感系统的协作。

图3则是一个更现实的场景，如果入口前面



图3　一个物流场站的入口

的车辆出故障、转弯半径太小等因素，导致车辆无法通行，我们就要在应用中设计独立的异常处理（Exception Handler），这不仅仅是程序设计问题，还需要系统的设计者考虑专门的物理通道。

所以，物联网应用的灵活性不只覆盖应用系统自身，还应该包括应用的物理场所等其他因素，这也是物联网应用区别于典型互联网应用、普通企业应用的地方。

可学习性

现在很多物联网应用中真实地存在一些问题，以至于用户真的“不知道”。

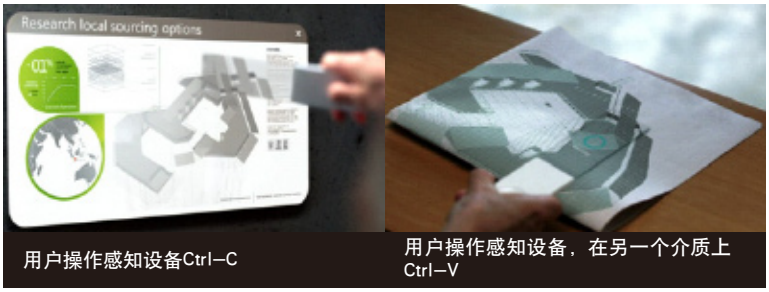
- 泛滥的缩写。诚然物联网是个较“新”的概念，但IT行业病往往喜欢大量使用这些缩写，试想作为普通用户，是写成“GSM M2M接入，2 RMB/M”还是写成“移动网每车每次通行收费0.2元”直观呢?
- 由于物联网环境下，预期用户面临的应用数量很大，因此随着固件和系统的升级，往往出现功能泛滥的情况，如何让用户学会他最用得到的功能也是个课题。

简言之，物联网应用也是应用，功能再多、再好也只是“心里美”，尤其在物联网时代，各类应用方案和竞争对手也将井喷，如何让自己开发的这捆物联网的“萝卜”从超市的的货架走到用户的菜篮子，还能卖个好价钱，要在可学习性方面下点工夫。



水杯显示温控信息　　“放下水杯”触发的更新事件，重新计算温控信息

图4　最少交互正例：通过拿起水杯触发温控提醒



用户操作感知设备Ctrl–C　　用户操作感知设备，在另一个介质上Ctrl–V

图5　最少交互反例：人工Ctrl–C、Ctrl–V

最少交互

前面已经擦边得提到过“最少交互”的内容，下面我们以微软Productivity Vision视频的截图看一个正例（图4）和一个反例（图5）。这段视频的用途不是介绍物联网，这里只是从“最少交互”角度作为参考，无其他任何直接、引申含义。

视频中这几个片段虽然不起眼，但恰恰能反映出“最少交互”部分谁的设计失分、谁的设计赚分。之所以认为后面的是“正例”，是因为它成功的关键在于如何用最自然的方式完成与用户的交互过程。类似的，在学校大门您是选择打卡上学还是采用防尾随快速通道也体现出设计师在“最小交互”部分的差别。

蕞少汜靛负擔

这个小标题不是印刷错误，而是“火星文”方式的“最少记忆负担”，图6中的两个设备前端对于不同用户而言，他们都是最少记忆负担的示例。



图6　最少记忆负担的操控前端

对于DJ而言，如果现场再去选择菜单调音，结果估计就是跑调；而对于一个功能复杂的手持定位系统，不妨采用简单的多层菜单缓解用户记忆负担。类似的情况出现在家里的越来越智能的电饭煲上，如果让用户选择“煮饭”—“流食”—“粥”的话也许不够人性，不如变成“米饭”、“粥”选得直接。

指导用户

作为泛联而且普遍存在的系统，物联网应

用要能够给予用户当时当处的指导。如图7所示，微软Productivity Vision的视频，指示接机路线时请不要给我展示一个T3航展楼的地图，我不关心，只要你告诉我从哪走、向哪个方向即可。



图7　直接指示接机路线

物联网应用中可能也需要“中文服务请按1、for English press 2”的碎碎念和植入广告，但对于深处物理现场尤其是较为危险现场工作的用户，简单、直接的指导更有用，其交互方式也要不拘一格，而不仅仅是屏幕右下角“砰”一声之后弹出个popup窗口，“讲给我”、“Show给我”也许更实用。

设计要点

上面我们谈到了用户体验@thing，下面谈下设计要点。我们需要在典型的应用分层中增加一个“体验管理”层，在不破坏现有应用层次的前提下，变MCD为UCD，如图8所示。



图8　增加“体验管理”后的物联网应用分层结构

根据项目经验，需要但不限于以下技术。

- 实时数据库：现场系统数量庞大，采样交互频率也更高，采用传统的商用关系数据库其性能、成本也许不合适，但为了实现“体验逻辑”管理又往往需要现场综合信息支持，因此实时数据库不失为一个选择。
- 工作流（H2H、H2T、T2T）：以往我们的工作流往往用于连接人和服务，但物联网还要增加对设备的协同，因为这些设备具有自己的收集、反馈机制，只有将现有的工作流机制扩展到设备，才能更便于用户使用，将用户置于物联网的中心，计算和处理围绕用户按需组合。
- 异常处理通道：物联网的异常处理除了代码逻辑外，还应考虑必要的物理通道，否则物联网就成了纸上谈兵的“假把式”了。
- 更全面的面向人的感知设备。

除此之外，安全问题是物联网面临的更大风险，网页中您感觉存在“钓鱼”的嫌疑，可以把它关闭。但随着物联网的广泛应用大量出现，也带来了一些法律问题和纠纷，相信这会成为律师们拉动经济增长的新增长点。您可能觉得用手指在笔记本上一划就能登录系统很Cool，但如果面临图9的情景，当您把手机放下马桶盖就从白色变成提示“老张不到10分钟又来了，欢迎您，上次您消费了3张纸，估计您还可能对另外3张纸感兴趣”，这样的情景是否多少会有些尴尬或不适应？



图9　具有智能感知和本地存储的马桶盖

小结

本文介绍了物联网应用设计中如何体现用户为中心的一些建议，但很肤浅，权作抛砖引玉。但有几个原则希望您把握：

它对于时效性的要求往往更高；

应用设计中要充分考虑@T或@H的物理空间需要和物理变化延时；

要尽量从后台协同@T、@H和既有的信息系统，而不是让人做网线和带宽连接他们；

物联网应用服务对象包括@T和@H，但归根结底服务的是@H。

（本文整理自作者在2010软件开发2.0大会上的演讲。）

务实的用户体验设计

——用户体验设计的2010年

■ 记者 / 董世晓



解研科

从事产品设计、用户体验方面的工作。现任职于百度，曾服务于搜狐、点击科技等公司。高度依赖互联网的资深用户，狂热的Web产品爱好者，坚信设计与互联网的有效结合将深刻地影响我们的生活。



西乔

《神秘的程序员们》漫画作者，Magicome公司创始人。新近创建designlol.net。从业经验5年，担任过Web设计师、项目经理、产品顾问等职位。



吴琼

多年从事设计工作，在建筑、软件、互联网等行业有丰富的经验。目前在Autodesk中国研究院负责核心产品部门用户体验设计团队的管理工作。

2010年有哪些印象深刻的事件？

解研科：我很早就开始做产品设计了，也就是很多人所说的PM。所以有关用户体验行业的事情，其实现在已经不是太熟悉，也很少参加相关的聚会或者专业会议，但我料想和前些年没有根本的变化。

2010 年好像没有特别的事件。但是我想说的是，如果设计的理想和产品不能很好地契合，而是单纯地追求艺术和完美，就不能独立地存在和发展。背离方向的发展，可能会导致产品南辕北辙。在某些时候、某种程度上，我是支持产品的专制的，设计虽然有感性的一面，但整体而言更倾向于理性。我希望更多有能力的设计师能够走进产品设计的领域，驾驭整个产品，把之前的知识和经验融入到自己的产品里。这时候你会发现大有不同。

以前做UI设计师时，思维有种本能的惯性，就是做一个很完美的界面，最合理的信息架构，最易用、最优雅的交互，能触动用户情感的视觉，这些追求没错，但如果过分追逐这些，并上升到职业理想的层面，就会有问题，容易陷入为设计而设计的境地。很多时候，我们需要平衡自己的精力和资源的投入，需要有效地影响结果。完美主义可以成为我们做事情的动力和激情的源泉，但是做事情还需要有理性的决断，相信很多开始投身产品设计的前UI设计师都有同感。

西乔：iPad和iPhone的上市大卖，移动用户的爆发性增长，LBS应用处处开花。和阅读有关的产品、需求成为热点：比如各类电子书，传统媒体的电子化，个性化自定制媒体的出现，一切都与人们使用方式的变化有关。

吴琼：新的产品和交互方式的涌现。比如iPad、iPhone 4的发布及其对交互方式的革命性改变；Kinect带来新的游戏体验；Windows Phone 7 视觉风格的变化，更重视内容。

在互联网和移动方面，有相当一部分依然是复制国外的模式，比如微博的爆发、对Foursquare的模仿。还有一些是同步的如团购这种新的消费模式。移动应用的发展势头迅猛，手机及平板设备上的游戏让大众乐此不疲。

技术方面，云计算的扩展和入门门槛的降低使更多的基于云端的应用变为可能，而Chrome OS/Web Store的发布充分体现Google“浏览器即操作系统”的理念。基于各种技术的平台随处可见，其上的二次开发或者应用开发给更多从业者提供了商机和发挥空间，例如针对建筑行业的Autodesk Revit BIM平台目前就有大量的二次开发需求。

2010年用户体验设计方面最大的特点和变化有哪些？

解研科：如果要说大的特点和变化，首先就是整个行业开始变得更务实了，至少以前认为UI就是王道的人越来越冷静了。少了些布道和概念理论，多了些更实际的实践和摸索。虽然整个行业的内容分享有点萎靡，但质量上，我认为慢慢有自己的内涵和体会了。

然后，对于整个行业而言，很多从业者开始转化，有的被淘汰，有的继续本分地做自己的UI设计（守得住寂寞，其实也很好，纯粹、深入），有的也和我一样开始实践到产品设计的第一线，而不是只在那用文字YY怎么做交互、怎么做UI了。毕竟，用户体验，还得通过

产品落地生花。单纯做UI的时候，经常很难理会其他角色为什么那么想，总是觉得很多事情没有话语权。其实不然，一切皆因为我们太单纯、太理想。

西乔：这一年UI设计方面的最大特点是更简洁、专注，注重模式，注重空间感和空白的运用。而最大的变化则包括以下方面。

- 手持便携设备的的使用模式和体验反过来影响了Web与桌面端等GUI的整体设计趋势。
- 即使是一些保守的网站设计也开始尝试富有诱惑力的设计选择。比如质量更高的图片、中心舞台布局的设计、制造冲突、加大文字排版中的对比等。
- 在产品设计上，精简灵活的专业化小型应用依然非常抢眼，它们的使用流程更专注和简单，在考虑用户体验的时候，不再是面面俱到，而是结合产品的运用现状和用户规模来有所取舍。许多不够完美却增长很快的轻型应用取得了成功。
- 交互设计趋向更有效和更低调，使用更自然的操作模式，提升使用乐趣。包括模仿一些与物理世界操作经验吻合的交互方式。
- 在功能性、规范性和用户体验之外，设计师开始寻找那些从未出现过的、能更深层次打动用户的东西——“一种在感情上赢得人们喜爱的东西”。所以更多有生命力、个性化的风格在GUI上出现。

吴琼：首先，用户体验成为行业的关键词。它不再仅仅只是设计师关心的事情，几乎所有的人都在谈论用户体验。这是相当好的征兆。用户体验绝不只是交互设计，而是包括了一个应用或者一个服务从头至尾的全过程。比如一个电子商务网站，从用户在线上开始浏览网页、搜索并选择商品、购买、支付、确认，到线下的客户支持、送货、打开包裹、使用商品等，所有与用户发生关联的行为都应该被纳入用户体验的范畴。

其次，用户体验设计从功能向情感的延伸。优秀的设计打动人的心灵，用户不应只是完成任务，而是在享受使用产品带来的愉悦。心理因素不可忽视。

再次，跨平台的用户体验设计。电脑、上网本、平板设备、手机等移动设备之间的界限渐渐模糊，一款应用在多平台上运行已经是普遍的现状。在这种情况下，后台的技术支撑和前台的交互设计的剥离使得跨平台的用户体验达到统一。

最后，用户体验设计与敏捷开发流程。敏捷开发应用越来越广泛，对市场需求有更快的反应。用户体验设计如何介入敏捷开发？设计师在敏捷团队中的角色定位如何？对原有的团队和流程有怎样的影响和改变？我们对这些问题有了越来越多的思考、探索和实践。

单纯从设计角度来看，还有以下一些特点。

- 简单易用至上。形式逐渐让位于内容。
- 输入形式的变化。在传统的键盘鼠标输入之外，Direct Manipulation，即对物体的直接操控的行为模式逐渐兴起。它使得用户与应用之间有更直观的交流。
- 隐喻式设计。模拟已有的行为模式，并在数字平台上复制，将人们的习惯自然地导入应用中。多点触摸技术使其变得简单易行。
- 个性化推荐。通过对单个用户行为习惯的数据收集和分析，及基于此的用户体验设计，向用户推荐其可能关注的产品和社会网络。

2010年最关注的技术有哪些？

解研科：以前定位自己是一个UI设计师时，可能比较关注技术的走向，现在反而不刻意关注了，因为有什么新技术，想不知道都不行。再说，现在的产品，技术驱动的实际上越来越少，体验驱动的倒是越来越多，关注技术不如更多地关注需求。

西乔：我比较关注的是CSS3、 HTML5和Web文本替换技术的成熟。这些新技术让设计师们可以去做一些以前想都不敢想的事情。当然兼容性的阻碍还很大。另外，数据可视化在国内也运用得越来越多了。

吴琼：我认为需求和技术永远是互相促进的关系，而设计是联系这两方面的纽带。技术对于设计来说不仅是实现创意的手段，更能激发灵感，为需求提供更新更好的解决方案。目前我关注的技术有HTML5、云计算、Multi-touch多点触摸、Augmented Reality（增强现实）、Adobe AIR、API等。

在实际工作中，面临的最大挑战是什么？这些挑战是怎样解决的？

解研科：最大的挑战倒还没有真正面临过。不过问题倒是不少，天天都有，接踵而至，永不停歇。其实说到这点，还是需要把一些细小的事情处理好。

保持开放的心态去学习，细致地感受生活和工作。比如你有没有注意到冰箱的储物层塑料片边缘的凹槽，那个设计是很有目的的。很多时候你需要设计师的眼睛、手和商人的大脑。

西乔：最大的挑战还是去战胜自己。要想在眼界和技术上有所突破，可以通过阅读、讨论等，当然最关键的还是自己动手去做。

吴琼：在这个开发占主导、产品经理定方向的行业里，用户体验设计师如何发挥更大的影响力，如何与项目团队其他成员更好地合作?

- 在一个产品开发团队中，设计师是用户的代言人。通过用户调查和研究，我们了解并理解用户的需求，并力求为这些需求寻找最佳解决方案。事实上，设计就是一个不断发现问题、分析问题并解决问题的过程。在人人强调用户体验的今天，设计师们必须站在用户角度与团队的其他成员沟通。毕竟一个产品成功的关键在于你为用户解决什么问题、提供什么价值。而用户也是付钱买产品的人。这就是为什么我们需要在设计开发过程中做用户测试。我们一直强调产品的易用性，而交互的简单往往意味着程序员更多的工作。从测试中得到的用户的客观数据，比设计师的主观论述更能说服开发团队。
- 相对比程序员，用户体验设计师和产品经理在某些企业中的界限会更模糊一些。产品经理需要对市场需求和趋势有总体的把握，而设计师更注重对用户心理和行为的研究。这两者之间其实是密不可分的。在Autodesk的一些产品团队，设计师在初期就与产品经理一起定义产品需求，通过从一开始就进行的用户研究和对设计原型的测试，设计师可以帮助纠正产品方向可能的偏差。尤其是在敏捷开发的环境下，设计师和用户之间持续迭代式的访问和测试，可以保证对需求了解的正确性，并随时做出相应调整。
- 对设计师的素质的要求。除了创造性这个对设计师最基本的要求外，沟通能力和销售能力同样重要。几乎没有这样的天才设计师，他的作品能让所有人倾倒，因为设计不是科学，答案没有唯一性，而几乎每个人都会对设计有所意见。在这种情况下，如何让其他人接受你的设计就非常关键。当然，产品最终评判者是用户。如果设计师的产品持续得到用户的认可，他自然会在团队中建立起威信。
- 最后，有支持用户体验设计的管理层。用户体验正在成为高管们的口头禅，这是一个好兆头。自上而下地推行以用户为中心的设计和开发一定比自下而上要容易。毕竟企业是以盈利为目的，而商业的成功与用户是不可分割的。

2010年有哪些有代表性的用户体验设计方面的文章和资源?

解研科：2010年读的东西很多，但如果要挑有代表性的，还真难，一心想当代表的文章很难是好文章，所以我还是比较喜欢轻松随意点的，能够带给你一丝的启发就很好了。另外，我建议每一个UX的同行，不要局限在一个圈子里，一样的思想和方法，那样彼此地看来看去，越看越美，因为共同点太多了。事实并非如此，我们需要擦亮眼睛看看，到底什么是对的、是好的、是合适的。以前我也容易犯这种错误，只看同行的文字，觉得其他角色都很傻、没品位、不懂用户。这些怪圈，需要靠自己一点点去摆脱。

最后，我建议设计师都去看看包益民的视频（http://www.findesign.cn/?p=3333），你要承认一些事实，并且尽早认识自己，找到自己合适的定位和方向，发挥自己的最大优势。当然，老老实实做设计，也没什么不好，只是希望心态更平和一点，毕竟我们本不是世上唯一重要的一群人。

西乔：我推荐的一些资源性文章如下。

- 50个最新的CSS技巧、教程和工具（http://www.smashingmagazine.com/2010/10//12/50-new-useful-css-techniques-tutorials-and-tools/）。
- 关于HTML5的经典资源——从入门到飞跃（http://sixrevisions.com/html/the-only-html5-resources-you-need-for-getting-up-to-speed/）。
- 移动开发工具包（http://www.noupe.com/tutorial/mobile-development-toolkit.html）。
- Photoshop终极工具箱（http://www.noupe.com/photoshop/the-ultimate-photoshop-toolbox.html）。
- 可用性测试神话 （http://www.alistapart.com/articles/the-myth-of-usability-testing/）。

吴琼：一直关注的网站有www.37signals.com、www.Ted.com、Autodesk UX Design Blog。

平时自己和团队会用到各类UI设计工具、prototype工具，如我们自己设计的card sorting工具以及各类控件库。www.surveymonkey.com是一个很好的在线用户调查工具。

用人类行为学理论指导Web设计

■ 文 / Alexander Dawson 译 / 王亚洲

人是理性动物，当我们浏览网站时，我们的大脑在不知不觉中作出一系列的决定，这些决定将影响我们的行动。这听上去并不稀奇。对所看到的事物进行推理的能力使得我们能够形成自己的决断，归结出正确的结论，作出正确的决定。如果我们不是每时每刻都在思考进而采取行动的话，我们浏览网页的行为就变成漫无目的地乱点鼠标。

行为心理学是一个不断发展的领域，了解该领域的一些知识可以让我们这些Web设计的从业者设计出可用性更好的网站。如果我们理解人的需求和情感的原理——人类如何解释眼之所见又如何对其采取行动——我们就能更好地理解用户，就能更好地设计网站的布局、字体排版和色彩。

这篇文章并不能代替一个专业的心理学学位，文章中提到的观点也不适用于所有情况，因为世界上没有两个人是完全一样的。如果能简单了解文章所列出的一些理论（心理学里只有理论，没有事实），你就能更好地理解你的网站在用户眼中的形象以及它是如何被使用的。

马斯洛需求层次理论

要让你网站的访问者感受到自我价值的实现，满足他们的自尊心。Facebook、YouTube、Twitter等许多社会性网站面临的重大挑战之一，就是如何让他们网站的访问者感到安全而且是受欢迎的。

一般网站的访问者大多都“潜水”，这种习惯已经存在多年，而且潜水者并没有减少的趋势。这些用户只是静静来到你的网站阅读内容而不贡献内容。如果你的目的是要建立一个社区，那么这种用户就会很让人沮丧。



图1 StumbleUpon鼓励用户彼此成为朋友和相互交流

一个有效的设计必须满足访问者的需求——虽然访问者的需求具体是什么还有待讨论，而且每个项目的目标用户的需求、用户各种需求的优先级也不尽相同。

关于人类需求的最有名的理论是心理学家马斯洛提出的“需求层次理论”。该理论指出了人类不同的需求及其优先级，有些需求比较基本，比如生理和安全需求，而最高级的需求则是自我实现。

基于他的理论，我总结出和Web设计相关的“网站用户需求层次理论”。下面按照重要性递减的顺序将它们列出。

可访问性：网站能够被所有人找到和使用。

稳定性：网站行为是一致且可信赖的。

可用性：网站是用户友好的。

可靠性：网站总是在线的，不会宕机。

功能性：网站提供用户想要的内容、工具或服务。

灵活性：网站可适应不同用户的需求和想法。

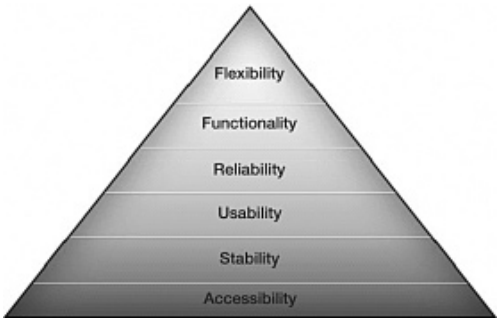


图2 网站用户需求层次

用户最基本的需求无非是随时可以访问网站、浏览网站内容、使用网站提供的服务和工具。

尽可能根据用户需求调整你的网站。如果网



译者简介：
王亚洲，目前就职于SAP中国研究院，从事用户体验、交互设计等方面工作，对Web设计、数据可视化及BI系统设计有所研究。

站不能满足上面提到的用户需求，尤其是基本需求，就会极大地影响网站的用户体验。设计师需要不断检查影响网站体验的元素，避免出现用户基本需求未被满足的情况。

美丽的诱惑（相貌偏见）

虽然视觉设计并不是一个网站的全部，也无法决定一个网站的生死（很多成功的网站有着平庸的视觉设计），但人类还是偏好美丽的事物。心理学家发现，人类对美好的事物的认知往往存有“偏见”，当然是好的偏见。

好的内容是网站成功最重要的因素，与视觉设计相比，网站的内容可以更加持久地影响用户，而且往往反映了网站的个性。

“相貌偏见”理论告诉我们一个漂亮的网站比一个长相平庸的网站更能吸引用户的注意。用户看到网站的第一印象如果是愉快和舒适的，他们就有可能会再次访问。

让用户感到愉悦是非常重要的，它是你的网站与用户建立长期关系的重要基础。出色的视觉设计往往可以帮助网站吸引初期用户的注意。当用户真正开始使用你的网站时，美观的重要性会降低，因为用户的关注点已经从表象转移到更深层次的东西，这些东西就是网站的内容本身。

不过也有例外，如果你的名气非常大，你可以把网站做得很难看，仍然会有大量的追随者来访问你的网站。这叫做“品牌认可”。比如可用性专家Jakob Nielsen的网站。

有人认为视觉设计并不能改变用户对网站的整体印象，但是“相貌偏见”理论告诉我们，虽然网站的内容对经常访问的用户是最重要的，但出色的视觉设计可以帮助你建立起初期的用户群体。

在不违反上节提到的需求层次的前提下，把网站做得尽量专业和美观就可以吸引很多偶然间访问到你的网站的用户（大多数用户被吸引都是凭借外表和第一印象）。一旦吸引住用户，网站上高质量的内容就有机会被用户发现了。

序列位置效应

信息的排列方式影响到人们记忆的效率。列表、图表和表格非常有效的原因，正是由于它们将大量信息打散成人们容易理解和掌握的小块。在大量使用文字的网站中，精心组织的信息块就显得很突出。

人类大脑认为列表开始和结束的地方更加重要，因此用户对列表开始和结束部分的信息印象相对深刻。



图3　Yahoo Answers使用小型内容块来帮助用户理解信息

Hermann Ebbinghaus提出的序列位置效应理论指出，人类记忆事物的准确程度和该事物在一系列事物中所处的位置有关。在Web设计领域，和这个理论关系最为密切的就是“视觉层次”设计。

一般情况下，人们在浏览网页时，只会拿出10%的精力去记忆网站上的信息（如果他正在寻找某个关键信息的用户，则只会拿出1%）。我们应该探索对用户有帮助的展现信息的方式，使得我们的网站和内容更容易被用户记住。

信息处理深度

信息处理深度（或信息处理层次）指的是人类要想记住特定信息时需要预先对其进行处理的强度。

可以通过事后对用户提问的方式来确保用户仔细阅读了网站上的重要信息。还记得我们在学校时那无休止的考试吗？虽然很多人都对考试心存畏惧，但是它对我们增强对信息的记忆是有作用的，这也是有科学依据的。

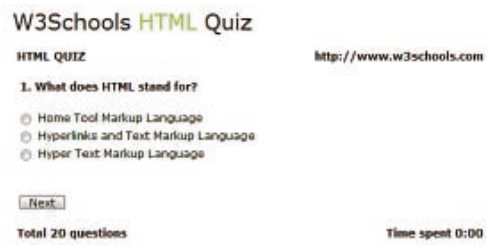


图4　W3Schools的每一个教程结束时会通过小测验的方式来加强用户对知识的掌握

通常人们不愿意被不得不回答的问题困扰，也不愿自己的浏览体验被这些问题打断，所以需要加入一些简单的机制来减轻用户的反感（比如在许可协议中的“我接受”按钮）。要时刻在告知用户和打扰用户之间做出权衡。

采用关键点提醒和适时回顾信息的方式来确

保你想传达的信息能传达给用户。例如，在网上购物结算时，通过让用户回答问题或重复地提示用户，来确保交易安全性。

如果你的网站是提供教育型的信息，简单的测试可以帮助用户记忆。甚至在博客文章结尾的一个简单直接的问题（比如，关于这个话题你有任何可以分享的吗？）都能鼓励用户去批判地思考他刚刚阅读过的内容。

在告知用户和打扰用户之间找到平衡点很难，设计师要根据自己的经验做出最好的判断。图片、交互元素、小测试甚至小笑话都可以被用来帮助用户记忆信息。

费兹定律

费兹定律（Fitt's Law）指出在人机交互过程中，移动到某一目标所需时间跟目标的距离与大小有关。在Web设计领域，与其相关的就是界面可用性设计。

网站的设计在很大程度上决定了用户实现其目标的难易程度。某些布局设计会阻碍用户目标的实现。一个经典的例子就是将需要点击的目标设计得非常小，以至于用户需要将鼠标精确定位到那一点才能完成点击。

费兹定律指出可点击的区域越小，激活它需要的时间越长，界面的可用性就越差。

在设计中要注意将网站上的交互区域尺寸做得足够大，注意利用触屏设备的交互优势，确保页面可以被缩放、字体能够被放大。减少用户对内容进行访问的阻碍，这对残障人士尤其重要。任何降低用户体验的元素都会使访问者很快离开你的网站。

认知负荷

手上的任务越多，完成单一任务用时就越长。认知负荷理论描述了我们的学习和解决问题的能力，是怎样随着需同时处理的任务的增多而下降的。

简单地说：我们给用户的任务越多，他们完成任务就越慢，而且感到越困惑。Steve Krug在他的《Don't Make Me Think》一书中采用“好感储存器”的方式描述了这个理论。

只有保持页面简单、结构清晰才能使用户更快的得到他们想要的信息。布局设计巧妙的网站是最易用的，也是最受好评的。你的网站应该让人使用起来毫不费力。

僵尸浏览效应

我们不是在谈论电影人物。“僵尸用户”是指在访问网站时有着明确的目标，不为网站上其他元素所干扰的用户。

用户对你的网站越熟悉，僵尸效应就越明显。就像逛超市，一旦你知道了你想买的商品的位置，你就不会四处乱找了。这也解释了为什么有些商店会经常改变商品的摆放——因为他们想让顾客看到更多的新商品。

你要的不光是一锤子买卖，你想让更多用户去探索你的网站进而增加收入。僵尸用户往往忽略周围的事物，仅仅根据他们的直觉去寻找信息。我称这种行为是不经思考的购买习惯！

克服僵尸效应很难。当然你不用像影片里那样重击他们的头，而是应该使用独特的设计去吸引和迎合用户。例如，精心布置的独特的元素会很容易得到用户的关注。



图5　在像Amazon这样的网站浏览时，我们的视线被引导向特定的商品

如果页面上所有的内容对你来说都重要，你就要决定哪个对你来说是更重要的 ——内容的排列可以引导流量。太多广告和堆积的内容会让僵尸们产生防御心理，他们会停止思考，忽略所见的任何信息。

单一的兴趣点更能吸引注意，如果它和用户目标相关，用户会有更大的可能去探索相关信息。允许你的用户根据自己的需要定制你的网站（像iGoogle和BBC）可以降低僵尸效应，因为用户会被与自己相关的内容吸引。

条件反射模型

在心理学中，条件反射指的是向个体灌输可预见性行为的过程。传统条件反射改变非自主行为，操作性条件反射改变选择（或改变个体做出某个选择的可能性）。这两种都和今天讨论的话题有联系。

想想用户条件反射似地关掉网站弹出窗口和关小网站背景音乐，这些行为几乎是出自本能的，是用户根据以往的浏览网站经验形成的条件反射。这些叫做“形成性行为”（Learned Behaviors）。

用户在浏览时不断遇到相同或相似的事物时，就会形成一些固定的行为模式。在Web设计中，一些设计约定、设计趋势和设计模式促使了用户将某些事件或界面元素与形成性行为相关联。因此，我们可以利用这些形成性行为去探索许多能够提高网站转换率的方法。



图6　对于“页面没找到”的形成性行为是点击浏览器的“后退”按钮

设计师应该关注如何改进网站的设计去适应用户的本能反应或形成性行为，而不是尝试去改变用户的行为模式。似曾相识的场景比全新的场景（比如用户面对网站布局的变化）需要更少的认知负荷去处理。在遇到全新场景时，用户的学习曲线会快速升高。这主要是因为要训练一个个体抑制自己的本能反应或形成新的本能反应是一个长期过程，往往需要许多年，而且并不值得。想要训练互联网用户的条件反射无异于逆水行舟。相反的，应多利用这些用户已形成的形成性行为。

让我们再看看操作性条件反射。操作性行为是针对用户的选择而不是反应。用户通过试错的方式在学习预测后果时形成特定的行为模式。

一个可以被用来影响操作性行为的方法叫做“引导”（Priming）。举个简单的例子：如果你明确地在网站上指出“需要帮助时请联系我们”，用户很可能会在想尽办法而无法达成目标时联系你。如果你没有这样指出，用户就很可能不会把联系你当作一个选择。如图7所示。

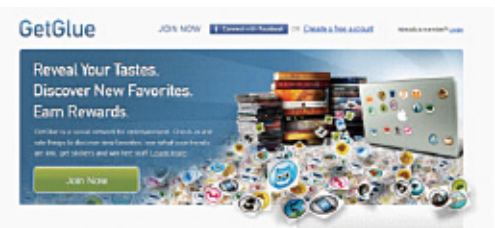


图7　通过展示重要的功能，Get Glue引导用户使用它的服务

操作性条件反射可以利用正面和负面的强化（奖惩机制）。这与Web设计有何关系？确保对网站有益的行为受到奖励（例如有用的用户评论），对网站有害的行为受到处罚（比如垃圾评论）。客户忠诚度本就很难获得，如果用户觉得没有受到公平对待，忠诚度自然会大大降低。所以公平地对待用户并且让用户知道。

如果谈到网站的功能，条件反射就变得很微妙，很难解释清楚。有激励人行动的口号的网站——比如让用户下载或者向用户索取信息的按钮——会引导用户去做相应的动作。如果用户发现结果还不错，往往会重复之前的行为。

条件反射的形成需要时间，而且人们如果感到有理由去遵守公平原则，他们就会去遵守。通过操作性条件反射，你可以改变用户对网站的看法，并且让网站更好地符合大多数用户的使用习惯（当然有的人会因种种原因反抗条件反射）。



图8　可口可乐使用正面强化：一个为客户制定的奖励机制

人性的复杂决定了我们要想完全理解它是非常困难的。时刻知道人们的行为是可以被改变的（通过恰当的刺激或满足特定的需求），对你的网上社区建设很关键。

总之，你不可能强迫用户使用你的网站，但你可以给他们足够的理由，给他们灌输足够的兴趣使得他们重复访问。社区里其他人的鼓励是非常有力的心理推动力。

结论

心理学在Web设计师从事的任何工作中都扮演着重要的角色。不管影响是外在的还是内在的，也不管你是否利用常见的设计趋势和形成性行为，理解用户的体验和认知原理对你来说都是一项巨大的优势。

本文译自知名Web设计师Alexander Dawson的文章，原文链接为http://sixrevisions.com/usabilityaccessibility/human-behavior-theories-that-can-be-applied-to-web-design。sixrevisions.com是为Web开发者和设计者提供信息支持的网站。

[感悟] 交互设计师的核心竞争力

■ 文 / 谢旭鸿

对于大多数人来讲，交互设计师仍然披着神秘的面纱。即便是和我们有紧密合作的工程师也不完全清楚我们在忙什么。他们需要了解什么是交互设计师：

- 交互设计是什么意思？
- 你们都具体做些什么？
- 你们的工作和我们有什么关系？

由于我的博客的关系，经常有一些学弟学妹给我写邮件请教。他们觉得自己有兴趣从事这个行业，但是痛苦于找不到入门之路，他们需要知道如何能够成为交互设计师：

- 我对交互设计很有兴趣，请问做交互设计师需要符合什么条件？
- 能够告诉我，我应该怎么做才能成为交互设计师吗？

交互设计师当然也是在现实的挫折、迷茫、振作和奋斗中，不断自我反思，不断追问自己，他们需要知道如何能够做好交互设计师：

- 我怎么样才能够感觉到真正在进步？
- 我有哪些努力的方向？
- 我如何能够提升自己的工作给予他人的价值？
- 我的核心竞争力是什么？

在前不久，和一位做管理的同事聊到对于交互设计师的招聘需求，以及季度考核指标的制定，也和做招聘的朋友聊过交互设计师由于“成就感缺失，感觉不到价值”流失的问题。他们需要知道如何去评价交互设计师：

- 期望交互设计师如何表现？
- 他们想要什么样的发展和提升？
- 如何去评价他们的表现呢？

本身作为交互设计师，我也希望能够给自己设定一个学习和进步的蓝图，帮助自己在这条专业路线上得到不断的提升。所以，我简单做了一些总结，仅做抛砖引玉，希望有更多的交流。

也拿冰山模型说说事儿

会线框图制作工具，了解人机交互基本原则和理论——能够帮助我们在人前侃侃而谈，但这只是做交互设计师的必要条件，而非充分条件。

很多新人在入门之初，将工具的重要性放到很大。但事实上决定我们能不能做好这份工作的因素并不在于使用什么工具。

著名的冰山模型告诉我们，人员个体素质的不同表现表式划分为表面的“冰山以上部分”和深藏的“冰山以下部分”。

其中，“冰山以上部分”包括基本知识、基本技能，是外在表现，是容易了解与测量的部分，相对而言也比较容易通过培训来改变和发展。而“冰山以下部分”包括社会角色、自我形象、特质和动机，是人内在的、难以测量的部分。它们不太容易通过外界的影响而得到改变，但却对人员的行为与表现起着关键性的作用。

那么，交互设计师的冰山模型会是如何呢？与“学来”相比，交互设计师更像是天生的或者培养成的？

我观察过一些不是做交互设计师的同事，他们在平时的工作中，表现出了很多适合做交



图1　交互设计师的冰山模型



作者简介：
谢旭鸿（Heidi），目前供职于阿里巴巴用户体验设计部，从事交互设计师工作。对交互设计、市场营销、产品规划、e-learning均抱有较浓厚兴趣。博客为http://heidixie.blog.sohu.com。

互设计师的特质，比如发送邮件时，特别强调一下人名和日期，考虑到大家没时间仔细阅读，灰掉了很多不重要的文字，将重点放大，让人在10秒钟内接受到重点——同时兼顾了那些时间短、想直接获得重点的人和那些想要获得更多信息的人的需求。他们如果掌握了一些交互设计师的工具、方法，我相信也许能够做出很优秀的交互设计。因为学工具和方法是相对容易的事情。

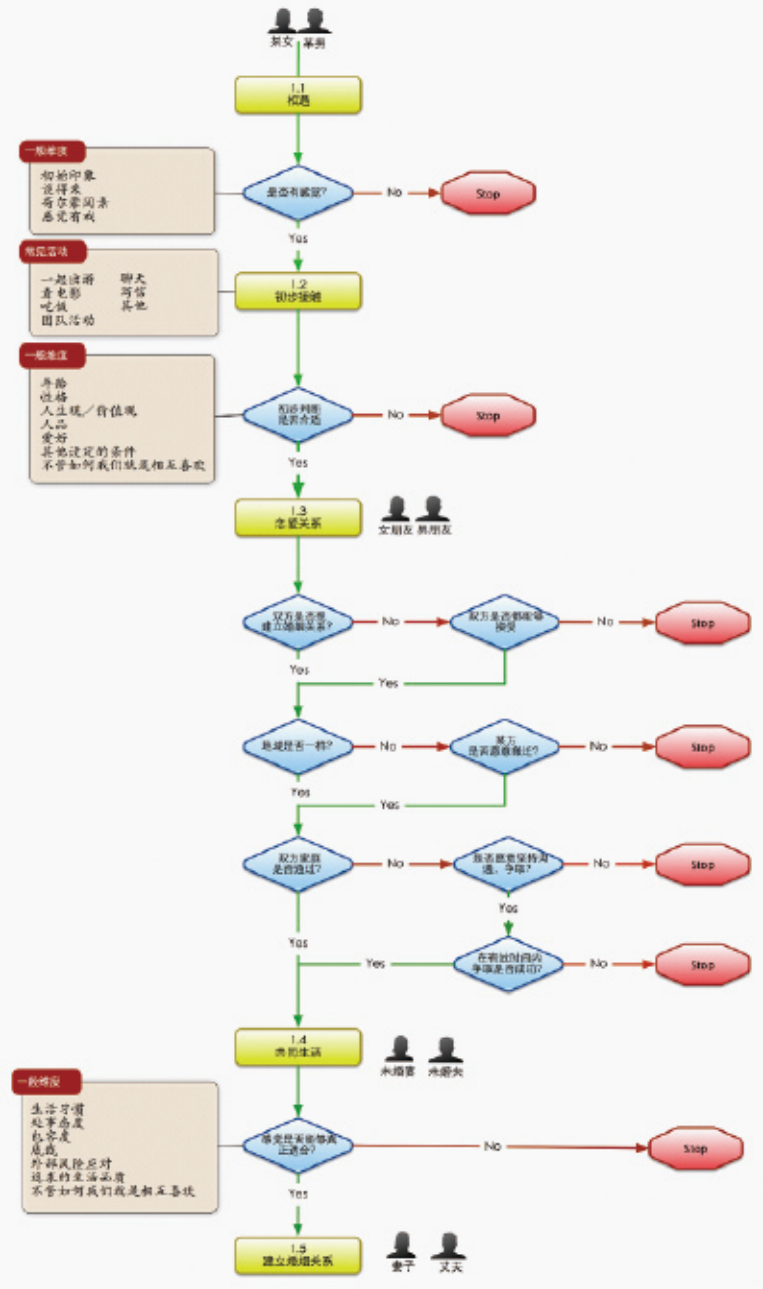


图2　感情流程图

但是以下的素质和习惯不是一两天就能够通过快速学习就获得的，这就如同习惯不能够习得，得在一定的环境中，不断培养和强化，最终让它们成为一种思维方式和工作习惯，带入生活和工作，成为你的一部分。比如：

- 以他人为中心，凡事的出发点都想想对方的反应和感受；
- 凭空想象复杂的交互行为的能力——第一步、第二步，想象每一步的可能性分支和风险；
- 纠结，越往细处做，就越纠结，一个流程分拆成几个页面，一个Button放到左边还是右边，一句文案对于用户行为可能会产生的影响，都需要再三掂量，在缺乏有力论据的前提下，交互设计师很多时候就会举棋不定，优柔寡断，甚至出尔反尔——第二天带着黑眼圈找前端工程师说，我想改一下昨天的交互方式，因为昨天做梦时发现那种方式有个缺陷。

如果觉得自己目前没有表现出以上特质的人，也不必担心，工具和方法虽然不是最重要的，但是它们也会影响你的做事方法，进而改变你的工作或者生活习惯。比如，当我做了信息架构图之后，我连搬家收拾东西，都需要先将我的所有物品罗列出来，进行分类整理。而经常思考流程图，导致有一次在帮朋友分析感情时，画出了一个感情流程图（在此仅供大家一笑），如图2所示。

从能做到做好，聚焦核心竞争力

知识、技能加上一些人格特质／内驱力和动机，三者相互影响运作，最终会形成一个交互设计师的核心竞争力。

核心竞争力是能够保证设计师专业性表现，在工作中创造出更大价值、取得更多信任的能力。

核心竞争力是能够帮我们不断实现从会做能做到做好的能力。

核心竞争力一定是一种综合素质，里面的分项相互影响，需要均衡发展，缺一不可。由于其综合性，所以成为稀缺；因为其稀缺，成为珍贵。这正是能够帮助我们从会做到做好，进而不断向优秀努力的能力指南。

一个人很专业是如何体现的呢？咱们给个通俗的定义：他通常能够把应该做的事情做到符合或超出期望。这个通俗的定义并不简单。

首先，必须得明确什么是该做的事情。一个交互设计师写代码很专业，也不能被认为是一个专业的交互设计师，哪怕是他自认为做交互应该写代码。因为这里“该做的事情”必须得是个人和企业形成一致。了解一下你的团队、合作伙伴、老板甚至业界对于这个职位的应该要做的事情的期望，这些期望是不是你想要满足或者认为自己需要满足的，如果达成一致，要先以满足此期望或超越此期望为基础。

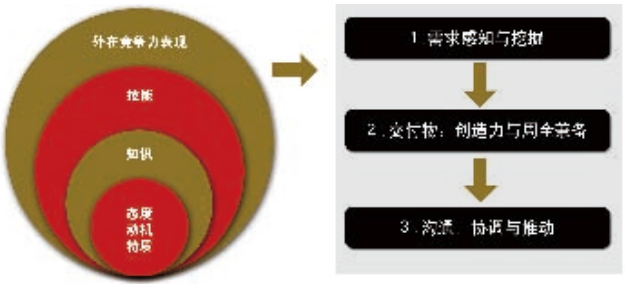


图3　交互设计师必须明确的该做的事情

感知与挖掘需求的能力

感知需求：利用交互设计师天生的敏感，主动地发现产品或网站的可以改进用户体验的机会点。

挖掘需求：不停留在任何问题和需求的表面，深入了解需求背后的意图。在日常工作里，甚至不必去关心反对意见本身，而是透过反对意见，去了解说者的目的和意图。

- 需求为王，问题的明确是解决方案的第一步。有人说，一旦明确了问题，其实就成功了一半。
- 谁更加能够感知用户需求，谁就能够讲出更加靠谱的可信的故事，谁就有说服力。
- 谁能够会更加站在商业需求的高度、站在战略的高度，谁就能够更加容易与商业方进行沟通，谁就更加容易赢得他们的信任，让他们更加放心把东西交给你做而不是由于不放心而指指点点。
- 谁能够挖掘到更深、更潜在的需求，谁就更加有可能超出期望，做出开拓性的、前瞻性的设计。
- 谁能够用用户的需求说话，谁就更加对自己的设计方案有信心，才会更加坚定去推进。

如何做呢？

- 多问几个为什么。
- 拓展信息收集渠道：尽可能找机会让自己和用户接触，并且记下他们对你的产品的任何一句反馈和建议，日后一定用得到。多多利用公司已有的调研材料和结果，尽可能多收集，想办法利用。
- 提高自己的数据分析能力——很多时候我们都以为数据分析就是看什么时候数据上升了、什么时候下降了，其实最重要的是从这些现象里挖掘出深层次的原因来。
- 多用自己的产品，模拟用户去用。
- 在以上的基础上，不断提高自己的信息收集、组织分析以及逻辑推理能力。交互设计工具箱里的mind map以及concept map可以帮助到我们。另外，来自市场营销学以及管理学的很多方法，比如swot分析、鱼骨图分析、竞品分析，都能够帮助我们提高分析能力。不然信息再多仍然只是信息，不能够提炼成为结论。

给出兼具创造性与周全的解决方案的能力

你可以说，感知/挖掘需求的能力我不如用户研究员，也不如公司专门的调研部门的同事，没错，他们比你更加容易得到调研、测试的资源，他们比你更加具备专业性。

但是你要比他们更加容易把拿到的调研结果，转化为清晰的用户需求，进而是解决方案。

解决方案，产品经理也能出。

视觉设计师也能出——事实上，任何人都能出。

但是作为交互设计师，你的要求是：要给出兼具创造性与周全性的解决方案。

- 尽可能想想符合用户习惯与心智但是有开创性、前瞻性的——时刻要加入点创意思考；
- 尽可能在满足大家（商业方+用户）期望的时候去想想如何超出期望——可用是最低要求，想用才是你的目标；
- 此外要尽可能去想想如何让方案具有更高的投入产出比。

如何做呢？

- 体验新产品，永远保持好奇心——有位同事做得就很不错，经常看他用一些很新奇的网站和产品，有时他会给你讲，我之所以做了这个方案，是受某某网站的启发，然后顺便分享一个很怪异的网站给我们。
- 收集，在网页上看到好玩的、有意思的设计，别忘记了截下来保存，并分门别类，慢

慢地，你脑子里会有很多参考对象。所谓的创意有时不是凭空想象出来的，而是被激发的。

- 提升自己凭空想象复杂的交互行为——每一种行为背后的需求，每一步操作可能会出现的情况以及应对措施，在把握大局的基础上细抠交互细节，交付出周全的设计（特别提醒，不要在躺下来的时候去想工作，不然你的梦境也许会充满了分支流程和各种交互控件）。
- 永远不要只给出一个解决方案。

沟通，协调，推动

没有这第三点，前两点做得再好，效果也大打折扣。很多新来的优秀的设计师，却往往会在很长的一段时间感觉到没有成就感。

其实第二点往往都做得超出期望，由于没有很多思维限制，方案一般都很有创造力。

之所以暗淡无光，其一，可能是因为第一点，业务不熟悉，对网站特定的用户缺乏了解；其二，不知道要找哪些关键的人，不知道怎么去讲自己的设计，不知道如何推动。

酒香也怕巷子深，在一家大公司里，噪声太多，干扰不少，大家的精力非常分散，你作为交互设计师，需要在一个短时间里找到合适的人，集中他们的精力，消除他们的矛盾（需求的矛盾），让他们对你的方案达成共识。这需要宣讲的能力、解决混乱场面的能力以及会议组织的能力。

在项目的推动过程中，你有时还必须要知道什么时候该找什么样的人来帮助你拿到结果。找对了，会让你事半功倍。有次，一位很优秀的产品经理和我分享他的经历，他说在他来公司的前三个月，就学会了一项核心技能：找人。知道遇到什么问题，该找什么人。有时找不对，就要绕很大的圈，还不一定能够解决问题，时间就白白浪费了。

怎么做呢？

- 清晰有力（说服力+吸引力）地去陈述设计。讲故事的能力很重要。交互设计师一个很重要的素质就是要“很诚信地讲更加可信的故事”。不靠YY，而靠观察、数据、案例分析，在讨论的时候，很自然用用户的场景去描述设计：用户来到这里，他带着什么样的需求；在遭遇到这种提示后，他会有什么样的反应；他的这种反应会导致他接下来怎么去做……要尽量像用户灵魂附体一样去讲。我见过一位优秀的设计师，在讲用户的使用场景时，虽然整个场景是他自己YY出来的，但是他的抓狂的动作、痛苦的表情，让人“感同身受”——到了这份上，想要PK产品经理，想要说服谁，一般都成功了。
- 要重视交付物，专业的、周全的、细致的、经得起挑战的交付物让你的说服力大增。
- 计划和时间管理能力，从参与项目开始，你需要全程（业务、设计、跟进、验收、用研……）参与，一直到发布后仍需要做数据跟踪，成熟专业的交互设计师不需要等待别人来催，他知道自己在什么时间节点上需要做什么、交付出什么，并且一般需要承担UED三个角色的接口人，类似于PM的角色，不但为自己的进度和质量负责，有时也要为视觉设计师和前端开发的进度和质量负责。
- 关注环境：有交际意识，关注别人的需要。你可能以后需要不少人帮你的忙。
- 思维能力——发散思维能够帮助你想出更有创造力的方案，积极思维能够保证你时刻充满斗志，时刻卷土重来。负面思维让你有风险意识，考虑到方案的风险和限制。逻辑思维帮助你理清思路，进行系统化思考，建立全局意识，还能帮助你组织材料，使之更有说服力。在大家争持不休，无法达成共识时，还能够帮助你解决混乱的局面，找到问题和争论的核心，帮助大家重新步入讨论的主线与正轨。
- 沟通能力——沟通，沟通，沟通。误会很多时候源于错误的沟通方式与态度。沟通的意义取决于对方的回应，如果对方误会你了，你需要重新阐述你的意思，而不是纠缠于对方的错误。
- 你越做越好，到了最后，就会形成影响力，影响力是非常奇妙的东西，让你的推动事半功倍。

受眼界、经验、个人能力限制，本文难免有些片面，在此仅作抛砖引玉，期望有更多的交流，一起提高。

多彩的产品之年

——产品经理一席谈

■ 记者 / 董世晓



王坚

糗事百科创始人，曾担任环球资源网站经理。2005年进入腾讯，先后担任QQ邮箱产品经理、Q吧产品总监、高级战略经理等职务。



杨武

网名品雪，思科系统中国研发有限公司软件工程师，同济大学MBA。近年来关注移动应用开发，主要作品有iCosta输入法、RockPlayer播放器。



高博

盛大创新院高级研究员，主管软件测试业务。曾在微软、惠普任职数年，出版了《C++：99个常见编程错误》、《微软的软件测试之道》等书。

2010年，都有哪些成功和失败的产品？原因是什么？

王坚：我认为Kik和Cut the Rope是今年比较成功的产品，游戏发展国也不错。彻底失败的没看到几个，有些表现不太好的产品现在还谈不上失败。

杨武：我觉得特别成功的是Apple的FaceTime。它用一组复杂的协议将视频通话变得极为简易，接近于One Click。其市场宣传从亲情入手，也抓住了要点，对比之下，移动营业厅里那个躲猫猫、装上班的手机视频通话广告简直是自曝其短。更重要的是，通过FaceTime，能够将iOS设备、Mac电脑整合成一个完整的在线视频服务平台，这就构成了完整的3屏方案，对于AppleTV可能是一个决定性的竞争优势。

另一个是微软的Kinect。虽然是以游戏周边设备的形式登场，但它是一款客厅信息交互方式的突破性产品。体感输入+配套的UI是TV从单纯的视频终端走向家庭互联网中心的关键。

我觉得比较失败的是Google TV。可以说，这是拿PC的思路套TV，没有很好地匹配人们在电视机前放松休闲的心理特点，操作时注意力需要在大屏幕和遥控器之前来回切换，同时没有拿出定制的信息输入方案，如果要用键盘，那直接连台电脑岂不是效果更好。

高博：2010年有很多成功的产品。其中最成功的产品我认为应该是以下几个。

- iPad，成功原因我想无须多言：仍然是将用户体验做到极致。苹果借助iPad的强劲销售业绩，市值一举超越微软巨头，气得鲍尔默临阵换将并准备亲自披挂上阵。
- Office 2010，它对“基础三大件”Word、Excel和PowerPoint都做了脱胎换骨的大改进，使其更完美地与云计算环境结合。而且，微软还推出了基于Windows Live Skydrive存储体系和Windows Live ID身份体系的Office WebApp，与Google Apps展开了正面竞争，并在数个月的时间内取得了决定性的形势扭转。
- 飞信，借助强大的研发实力和免费短信营销策略，成功地跻身于一线即时通信软件行列。这在有强大的腾讯QQ和Windows Live Messenger的寡头竞争市场压力下，是非常不容易做到的事。现在客观地说，飞信的体验已经好于Windows Live Messenger，仅仅在文件传输

等领域与QQ相比稍逊一筹，但是已经对后者形成了非常明显的竞争压力，因为只有飞信是真正依托现实社会关系的即时通信软件，借助中国移动的物联网基础设施和战略布局，其可以发散的方向几乎是无限的。

● 盛大Bambook和云中书城服务，真正的图书电子化落地平台设备和服务一体化产品，而不是“变薄了一点，并改名为电子书的MP4”。它完全以盛大文学的内容联盟网站为依托，不依靠设备盈利，因而不仅在同等价位的设备中提供了最优质的硬件和设计，更是新一代出版形态的代表产品。

而失败的产品，我认为有以下几个：

● Bing搜索引擎。微软缺乏互联网产品基因的劣势再次暴露无遗，尽管和Facebook已经达成了合作协议，算是一大资源优势，但是Bing的搜索结果仍然是一如既往地令人失望——不要说中文，英文亦是如此。看来Bing想要具备和Google竞争的实力，不知道应该说“路还很长”，还是应该说“距离只会越拉越大”。

● Google Wave网络服务。真的是因为这个产品“概念太超前”吗？我不这样看。因为Gmail同样是概念超前的产品，比如Ajax界面和超大存储，却一夜之间席卷了电子邮件市场。体验差、质量不过关，这应该是Google无可奈何地宣布这个产品下线的所有原因，也是无可推卸的责任。

当然，还有一些失败产品并不是由于它们的品质，而是由于它们的目的。比如360和腾讯互相拆台时开发的一系列软件产品，技术含量不可谓不高，但是它们却曲解了软件产品的意义，在用户的平台上摆开了私怨的战场，受到了相关部委的通报批评，这场闹剧才是真正的失败！我们软件产品经理应该引此为戒，严守道德良知的底线。

请从国内产品经理的角度，谈谈产品研发、管理、运营方面的趋势。

王坚：我感觉近来大的趋势是产品文化的建设。大家首先要有相同的价值观和理念，在一个强文化氛围之下对具体的工作方法进行探索。前几年更多的还是实践一些具体的工作方法，结果发现方法虽然能学，但是没有文化支撑，执行过程中就变形了，把文化课补好，比照搬一些具体的方法更重要。

杨武：在这个问题上，我想说的是，产品研发的整个趋势是更为灵活和快速。

高博：2010年可以说是互联网公司的桌面应用产品之年。8年前，我们从桌面纷纷奔向Web；8年后，我们又以同样的规模和效率从Web奔回桌面。正应了那句话“七八年就来一次”！不过，桌面的概念经过这次否定之否定的洗礼之后，已经发生了深刻的变革。桌面已经不仅仅指“PC的桌面”，而是已经拓展到了手机和平板电脑等移动计算设备上。相应地，桌面应用产品也已经如影随形地拓展到了这些疆域。但是真正深刻的变革在于，经过多年的Web革命以后，应用程序提供商终于意识到三个非常关键的问题。

第一，Web应用只是桌面应用的一个子类，因为浏览器本身也仅仅是一个桌面的应用。

第二，人们已经逐渐厌倦了一成不变的浏览器那层极像外壳的枷锁，而真正的桌面应用则可以提供Web应用所完全不能够提供的友好界面和复杂互动，并能够最大化地利用设备的本地资源。

第三，也是最重要的一点，那就是用户切换Web应用程序的成本简直为零，其区别不过是在浏览器的地址中输入a.com还是b.com，而用户对于桌面应用的惯性和黏性则是千百倍的。人们往往会说“这个程序我用惯了，不想再用别的了”。所以，先发制人地占领用户的桌面，竟然成为了互联网公司的竞争热点，这不可思议的表面下隐藏着非常实务的逻辑：占领了用户的桌面，就基本上等于排他性地占领了自己提供的互联网服务的领地。互联网并不等于Web，信息可以存储在云中，但是用户真正每天在用的，还是他的设备以及设备上的桌面。

桌面应用产品的战争序幕是由搜狐发起的，作为一家原先完全没有桌面应用产品的公司，它却利用自己的搜狗品牌率先推出了PC输入法、手机输入法和浏览器产品。仿佛巨石投入死潭，其他互联网公司才如梦方醒：它们缺少的不是对自己原先服务的进一步增强，而是对于一般桌面产品的短板。于是，在短短不到一年的时间内，腾讯从即时通信产品发力、360和金山从安全产品发力、迅雷从下载产品发力，纷纷推出全线桌面产品，其中白热化的产

品乃是桌面软件管理产品以及浏览器产品。想想也并不奇怪：控制了桌面软件管理产品相当于控制了桌面软件的入口，而控制了浏览器就相当于控制了Web的入口，看来这些互联网公司哪一个都不傻呀。

现在盛大也面临同样的问题，我要说，盛大的桌面产品已经蓄势待发，一定会有精彩表现。先发制人固然强势，但笑到最后才笑得最好。

如何看待国内产品经理这个职业？

王坚：大公司还好一些，真的是在做产品经理。中小公司授权不足，老板是真正意义上的产品经理，所谓产品经理只是产品助理。

杨武：国内对于产品经理作用的认识正在好转，大家越来越认识到产品经理对一个产品在市场竞争中的影响是巨大的。可以说两家公司在同质产品上的竞争，本质上是两家公司的产品经理的竞争。产品经理的眼界是否宽广、眼光是否长远非常重要。

高博：国内产品经理很辛苦，小公司的产品经理缺乏有实力的研发团队支撑，而大公司的产品经理又被烦琐的流程和“研发牛人”的骄傲所拖累。产品经理，最需要的是一颗平常心和换位思考的能力。

实际工作中，您面临的最大挑战是什么？这些挑战是怎样解决的？

王坚：我面临比较大的挑战是创建一个好的文化氛围，资金是否充裕、行业内的大环境是否与我们想要的小环境一致等方面都有很多挑战。目前还在摸索中，没有一个比较完善、系统的解决方案。

杨武：目前我所面临的最大挑战是怎样确定产品的差异化竞争优势。要想解决这个挑战，就我个人感受来说，建立一个远大的目标，用来指导我们对市场和用户的系统化思考，效果还是不错的。

高博：我面临的最大挑战是：新产品和新平台带来的用户需求的巨大变动引发的挑战。我的解决方法是，将新需求分解为“真正的创新核心”和“旧瓶装新酒”两个部分区别对待，其实表面上完全新的东西，往往大部分仍然是建立在旧有的产品和服务之上的，有很多经验可以稍加改造就加以应用。而对于真正的创新核心，则不要被它有时和旧的表现形式的相似性所迷惑，大胆扬弃旧思想，从小学生当起，进行学习和研究。掌握了它的规律以后，就不难搞清楚用户的需求变动趋势，从而在产品中及时、到位地响应它们了。



2010年有哪些有代表性的关于产品方面的文章和资源？

王坚：举贤不避亲，推荐我写的《结网@改变世界的互联网产品经理（第二版）》，内容有大量更新。

杨武：前些日子《花花公子》重刊了1985年对Steve Jobs的访问，非常有启发意义。

当我们拥有了触摸屏、移动宽带、云端资源来设计产品时，跟Steve Jobs当初面对鼠标和图形用户界面时的兴奋与困惑是很相似的。

高博：我推荐的图书有《人本界面》，Jef Raskin著；《软件创新之路：冲破高科技营造的牢笼》，Alan Cooper著。

我也推荐一本自己翻译的书：《设计原本——计算机科学巨匠Frederick P. Brooks的思考》。

我们到底是不是产品经理

■ 文 / 苏杰



作者简介：

苏杰，浙江大学硕士，2006年毕业加入阿里巴巴集团，一直担任产品经理至今。现在主要负责产品的战略规划、业务架构、数据分析、用户体验等，并于2009年开始负责公司内产品经理入门的培训。从2007年开始，每周记录工作中的体会，积累近20万字，整理大半年作成《人人都是产品经理》一书。博客地址为http://iamsujie.com。

我从事的是IT行业，具体点说，是互联网和软件行业。很多人可能都会和我一样，入行久了，看了一些资料，发现里面说的产品经理和我们平时做的事情似乎不大一样，于是做得越久越迷茫，总在问自己——我们到底是不是产品经理？

有这样的疑问很正常，因为有“产品经理”这个词的时候，还没有我们现在熟悉的“互联网”和“软件”的概念呢。从互联网、软件行业巨头的“产品经理”招聘广告中，就可以发现这个职位的内涵和以往的产品经理已经有着很大的不同。那么，互联网、软件行业的产品经理在概念上究竟有了哪些变化，有了哪些发展？为什么会有这些变化和发展？这会导致产品经理的职责、技能要求有哪些不同？下面我试着给出自己的认识和分析。

需要说明的是，我会经常把互联网和软件业的产品经理放在一起讲，这是因为两者同属IT行业，而且现在的互联网产品越来越复杂，越来越像软件，而软件产品也越来越多地基于网页浏览器，从产品经理的职责、技能角度来看，两个分支领域日益融合，越来越趋同。

产品究竟是什么

我特别喜欢想一些貌似“终极”的问题，这次的问题是：**产品究竟是什么**？按照我的理解：**产品就是用来解决某个问题的东西**。

现在我对着电脑屏幕敲下这行字，用的键盘是一个产品，显示器是一个产品，显示器里的文字处理软件是一个产品，我用的输入法软件是一个产品……忽然我口渴了，手边的杯子是一个产品，杯子里的果汁也是一个产品，我想到了买果汁的时候导购员提供的参考意见很有帮助，她的服务也是一种产品……最后，我觉得我自己也是一个产品。

所以说，产品这个东西，可以是有形的实物，也可以是无形的服务，多种多样。而解决问题其实就意味着满足人们的需求，这样才能产生价值。这个价值不仅要给产品的使用者，也要给产品的创造者。这本书里谈到的主要是一类产品——商品，并不是所有的产品都要变成商品，公益性、非盈利的产品也随处可见。但我们工作中所做的产品，绝大多数都是在人们的需求，即用户目标和公司的商业目标之间寻找平衡。只考虑用户，公司无法盈利，必然死掉；只考虑商业，光想着公司得好处，用户留不住，公司也会死掉。

所以在这本书里我们说产品是什么，产品就是要同时解决用户的问题和公司的问题，一个都不能少！

产品经理横空出世

在商品出现了很多年之后，产品经理的概念才第一次在美国的宝洁公司出现。此前，产品经理要做的事情显然也是有人做的，为什么这么晚才有人蹦出来说：“要有专人对这个东西负责？”

可以说，产品经理的出现是为了适应公司发展的需要。随着企业越来越大，产品越来越多，越来越复杂，原来按职能划分部门的组织结构已经无法适应，所以出现了产品管理的矩阵型组织，而此时产品经理的主要职责是规划产品的生命周期，负责产品的上市策略、定价策略、整合营销策略、销售与分销策略，等等。随着产品管理体系的运行，我们发现它还有很多好处，比如鼓励了创新、更重视用户等，这些话题以后再和大家仔细交流。

上述框架下的产品经理，我把其称之为“传统意义下的产品经理”，美国的琳达·哥乔斯在《产品经理的第一本书》与《产品经理的第二本书》中（注：原名是《The Product Manager’s Handbook》），提到了上述宝洁的例子，也有更详细的关于产品经理职责与技

能要求的表述，和以往不少书名中包含“产品经理”的书籍一样，总体感觉它们都偏重于讲述一个产品已经做好以后，应该怎样管理，怎样营销，偏重于市场、商业端，这类产品经理也可以叫做“品牌经理”或“产品市场经理”。

他们真是招产品经理吗

说到这里，也许在互联网、软件公司中做产品经理的朋友要跳起来了：好像之前产品经理要做的那些事情已经另有分工，由运营部门、市场部门的同事负责啊？通过调查，我们发现互联网、软件行业的产品经理确实和传统行业的产品经理不同，而各家公司对产品经理的招聘要求大同小异，“产品经理”这个概念，确实已经和旧有的产品经理概念不一样了。它更多地侧重产品本身“从无到有”、“从有到优”的过程，更多地涉及了“产品规划、数据分析、用户研究、需求分析、功能设计、项目管理、敏捷方法”等内容，而不是如传统的产品经理那样，去做已经有了产品之后需要做的诸如管理产品、推广和营销产品的事情。

产品经理概念的进化

是我们把产品经理的概念理解错了，还是产品经理的概念变了？事实表明，在互联网、软件行业，业内称为“产品经理”的，90%都不是传统的那个概念了。没必要纠缠于名词的解释，作为奉行实用主义的职场人士，我们不妨接受事实，就继续使用“产品经理”这个词吧！

所以，我们有必要来讨论一下互联网、软件的产品经理与传统意义下的产品经理为什么会有这些差异？而这些差异又让从业者需要担负哪些不同的职责，提出了哪些不同的技能要求？下面抛砖引玉谈五点看法，如表1所示，值得一提的是，我说的对比都是就整体情况而言，相信会有特例。

行业形态不同：成熟行业VS.新兴行业

传统行业经过几十年乃至上百年的摸爬滚打，市场已经成熟，产品基本定型，通常只有渐变式的创新，很难有重大突破。另一方面，用户也已经成熟，对产品相当熟悉，也已经形成比较固定的使用习惯，较难改变。所以对于这样的市场和用户，公司会偏重营销类创新，可以参考《公司进化论》（全名为《公司进化论：伟大的企业如何持续创新》）里的描述，映射到产品经理的职责上也就自然而然是琳达·哥乔斯的书中所述的那样。

而互联网、软件行业是新兴行业，三天一小变、五天一大变，产品本身在不断取得突破，用户看什么都是新的，所以产品需要推陈出新，尽力先入为主，占领用户，主导用户习惯，这就导致了产品工作的重头戏在前期，从无到有，从有到优，偏重研发类创新。

因此，互联网、软件行业的产品经理更重视产品功能本身的规划，需要“对市场发展趋势有敏锐的洞察力和创新意识及良好的分析、研判能力”，要能不断改进产品，要“深入了解业务，挖掘用户的多种需求，不断推出有竞争力的产品”，“制定所负责产品线的发展蓝图和实施路线图”。

产品形态与成本结构不同：实物VS.虚拟物品

传统行业的产品多为实物，所以有采购、仓储、物流等分工，产品研发出来以后，还有大量的制造成本，这也使得传统行业的产品经理有相当多的工作是需要考虑如何把整个供应链打通，怎样销售、分销、促销，等等。而互联网、软件产品多为虚拟物品，公司相对而言显得较“轻”，不管是团队，还是成本花费，都更加集

表1　产品经理概念的进化

	典型的传统行业	互联网、软件行业
行业形态	成熟行业	新兴行业
产品形态与成本结构	实物	虚拟物品
生命周期	几年	几个月
盈利模式	单一卖产品赚钱	多元盈利
用户心态	花钱买	免费用

中在产品研发的过程中。

一个常见的互联网产品，很可能只是由几个人或十几个人、几十个人的团队所做，而用户却是上百万、上千万，甚至亿万量级，这在传统行业不可想象。虚拟物品的复制成本极低，所以重点资源会投入在产品本身，较少考虑实体经济里供应链上下游的事情。于是，对产品经理来说，需求分析、设计的细节尤为重要，必须亲自把握，可能一个细节的改进就能增加上万的用户，杠杆效应也十分明显——因此，在招聘的广告词里自然就有“善利用设计工具完成产品用例设计和Demo 制作”这样的要求。

生命周期不同：几年VS.几个月

传统行业产品典型的研发生命周期一般是几年，甚至更长，所以需要比较复杂精细的流程来支撑，比如我了解到在汽车行业，做一款新车的整个过程中，有不下300个评审点，这样复杂的过程显然必须由经过专门训练的专人负责。而互联网、软件产品典型的研发生命周期就只有几个月，所以研发管理过程会更精简，一个典型的产品研发过程，一般只有10 个不到的评审点。

于是，我们推崇敏捷方法，传统行业的精雕细琢不适用了，我们要快，这也是为了顺应新兴市场的需要，而且船小好掉头，有问题也能够快速地改，再发布一次升级就行，不像传统行业改起来那么麻烦，问题产品只能“召回”……所以，虽然互联网、软件的项目更加不可控，但项目过程本身看起来并没有传统行业那么复杂，更接近“艺术”而不是“技术”，要依靠丰富的经验，于是产品经理也经常兼顾项目管理，这样自己可以在项目完成度和产品质量之间做平衡，对产品无疑也是一件好事。所以我们在招聘广告中看到公司要求产品经理“发起产品研发项目”，“组织资源实施产品，对其效益负责”。

盈利模式不同：单一卖产品赚钱VS.多元盈利

传统行业的盈利模式多为通过卖产品赚钱，或是直销，或是通过渠道分销，总之是靠产品本身的价值来赚取利润，而互联网、软件产品有很多产品本身是免费的，一部分产品甚至几年内都不考虑赚钱的问题，可想而知，对做这种产品的产品经理，要求肯定不同。而另外一些能赚钱的产品，也有着更多元的盈利模式，比如免费给用户使用，但利用用户的注意力赚取第三方的广告费等。

盈利模式的差异造成了产品为谁做的差异，传统行业多是为付钱的客户做，值得注意的是，很多情况下客户只是买产品的人并不是用产品的人，也许搞定几个大客户，就3 年不愁吃穿。

而互联网、软件产品大多是为使用产品的终端用户所做，而且通常是面对海量的用户，用户数多了，自然就能盈利。所以，互联网、软件行业的产品经理会更重视用户研究、数据分析等工作，要“负责用户研究，把握用户需求，实现用户需求”，而盈利的事情反倒不用直接去管，会有另外的团队负责。

用户心态不同：花钱买VS.免费用

传统行业产品的用户知道，东西是买来的，花了钱的，所以有点不爽的地方也就凑合着用，不至于把产品扔了立刻去再买个新的。而互联网、软件产品就不同，大多数都是免费的，每类产品雷同的还很多，所以只要这个产品用得稍稍有点不爽，用户马上就能很方便地找到另外一个试试。

于是，互联网、软件产品更重视用户体验，相应的，出现了很多产品经理会涉及的工作内容，如交互设计、视觉设计、文案设计等。举个例子，有时候为了确定两个按钮是上下分布好，还是左右分布好，我们有可能做大量的用户实验。在互联网、软件行业中，产品经理能真正体会到“用户是上帝”的感觉，辛辛苦苦做一个产品，给用户免费用，还要尽量让用户免费用得比付费的都爽。

小结一下，以上五点相互之间也是紧密联系、相辅相成的，共同造就了传统行业与互联网、软件行业的产品经理的差异。比如为了给用户极致的体验，我们需要做很多数据分析，而数据分析的基础在于互联网、软件产品的虚拟特性，可以大量地记录用户的各种行为数据，这是传统行业很难做到的。又如正因为是新兴市场，所以产品不成熟、用户不成熟，于是产品的生命周期缩短，需求变化快，项目中不可控的因素增多，使得敏捷方法备受推崇。

非典型产品经理

讲了半天，可以看到，尽管都叫产品经理，但我们这里所说的互联网、软件业的产品经理，

他们所做的事情和以往传统行业的产品经理相比，已经发生了很大变化。不过，你可能还有一个疑惑和一丝忐忑——平时我只做上述谈及的一部分工作，这样也能叫产品经理吗？

我的回答是：能。这是因为，不单是传统意义下的产品经理，就算是新概念下的产品经理的职责，也通常分给几个人或几个部门做。很少出现全能型产品经理，就算全能，也会因为个人精力所限，在某个时间段只会专注某方面的工作，或者只能是对每个方面都蜻蜓点水，而倘若如此，那么这样的产品经理多半只能是事业部的总经理，或者公司的CEO。于是，我们很多人都成了新概念下的非典型产品经理。

这一点在传统产品经理的框架下已有线索：哥乔斯早在其著作《产品经理的第一本书》的最后一章“产品管理终将走到尽头？”里谈到各界对产品经理、产品管理制度的质疑，提出产品经理制度仍需要不断修正，并指明了三大变化方向：**产品管理团队、事业单位经理的任用、更专业取向**。其中，产品管理团队顾名思义是用团队的力量来代替单一的产品经理；而任用事业单位经理的制度则是从项目管理团队的概念发展出来的，把产品强调为一个事业单位，也就是我们经常说的事业部，而产品经理也就摇身一变成为了事业部的总经理，区别在于总经理有了更大的权力；更专业取向则可以用下面这段摘录的文字说明，这也比较像我个人的一段经历：

由于企业分派给产品经理的责任愈来愈多，有的公司开始质疑——对产品经理一个人来说，这样的负担是不是已经难以负荷。结果造成现在出现缩减产品经理的责任，让他更为专业化的趋势。至于要求产品经理专注的方向，则因公司或行业类别而不同……

有时高科技企业会采取类似的做法，让产品经理专心处理产品在工程和技术方面的问题，而把大部分营销决定交给另外的职能单位来负责。在这种情况下，产品经理可能会变成产品技术/应用方面的专家，他最主要的工作就是支援、协助销售人员——至于了解市场和从市场了解产品利益等工作，则另有他人代劳。像这样把营销功能和产品开发活动分开来处理的做法也有风险：产品经理将失去和顾客间的联系，而且会因为对产品太过熟悉，以致丧失判断上的客观性。

不管企业想用怎样的专业取向，以便产品经理更容易地管理他的工作，很重要的一点是，请记住这个职位当初是为何而设——想要更了解产品与它面临的竞争情况，最终目的是要满足顾客的需求。

不管整天只是写文档、做Demo，还是成了没钱没权的项目经理；或是求完销售求工程师，最后只能欺负客服，到处不招人待见……前辈们似乎也认可——人人都是产品经理，非典型成了多数，也就变成典型。

一线员工眼中的管理

既然产品经理这个职位中有个“经理”二字，似乎多少就有点管理的味道。

管理的能力，其实就是“在资源不足的情况下把事情做成”的能力，这里的资源在产品经理的工作中通常表现为以下几种形式。

第一，信息不足以决策。时间有限，能力有限，每次决策前不可能掌握所有信息，做决定时总是很头疼，我估计是“拍脑袋”拍得太多的原因。

第二，时间不足以安排周密的计划。总是接到3 个月、1 个月，甚至1 个礼拜完成某项目的命令，每次都让我们张大嘴巴说不出话来，应承下来后如何计划？不过一次又一次的实践表明，办法总比困难多。

第三，人员不足以支持工作强度和难度。不但时间不足，人员也不足，就算数量足，能力够不够？能力够了，团队士气高不高？哪个公司不加班，又有多少公司有加班工资？但还得完成任务，难不难？难！

第四，资金不足以自由调配。俗话说钱要花在刀刃上，买机器要钱，招人要钱，产品推广要钱，而花这些钱的前提是公司还得赚钱，每一分钱都恨不得掰成两半用。

以上四点还可以推广到生活中的各方各面，凡是资源，**总归不足——这是常态**！既然不足，就需要学会分配资源、管理资源。比如说自己的时间、衣橱、工资……其实，你已经每天都在做了，不是吗？所以你已经是产品经理。

（本文节选自电子工业出版社博文视点（武汉）出版的《人人都是产品经理》一书，在此感谢博文视点的授权。）

TOP10 2010年度CSDN十大资讯（编辑推荐）

MIT技术评论：2010十大新兴技术

概要：本文列出了《MIT技术评论》评选出的2010年度十大新兴技术，其中四项新兴技术实时搜索、社会化电视、移动3D、云编程均属于IT领域，由此我们可以一窥未来IT领域技术的最新进展。

链接：http://news.csdn.net/a/20100426/218108.html

Google的十个核心技术

概要：该文列出了Google的十大核心技术（GFS、Chubby、Protocol Buffer、MapReduce、Sawzall、BigTable、数据库Sharding、数据中心高温化、12V电池和服务器整合），并对这些技术的架构进行了详细的分析。一句话，了解Google技术难得的资料，值得珍藏的技术佳文。

链接：http://java.csdn.net/a/20100813/278148.html

【观点】风雨20年：我所积累的20条编程经验

概要：文中分享了DCS公司的创始人Danylko在20年编程生涯中总结出的20条编程经验（比如估算解决问题所需要的时间、不要过于注重程序的“设计模式”经常备份代码等），涵盖了编程心态、编程方法、工程管理等多个方面。这些经验看似简单但要做好实非易事。

链接：http://news.csdn.net/a/20101113/281933.html

2010最危险的编程错误

概要：本文从CWS（美国国土安全部下属的软件保证项目）与SANS（权威安全培训组织）两个机构联合发布的2010年编程错误列表中摘录了其中最新、最危险的25条（比如跨站点脚本攻击、SQL注入、经典缓冲区溢出等），并与去年相比，指出今年落榜错误名单仍存在复发风险。

链接：http://news.csdn.net/a/20100219/217067.html

摩根士丹利7月互联网趋势报告：腾讯创新能力超微软

概要：本文公布了摩根士丹利公司发布的一份“互联网趋势”报告，称腾讯在创新能力上超越微软，仅次于苹果、谷歌和亚马逊，位居第四。读者就“腾讯创新”展开了激烈讨论，评论达487条之多，部分网友对腾讯所取得的成功表示支持。

链接：http://news.csdn.net/a/20100613/218801.html

软件架构师应该知道的97件事

概要：本文摘录了《软件架构师应该知道的97件事》一书中提到的97件事，比如客户需求重于个人简历、关键问题可能不是出在技术上、架构决定性能等。这些经验是对海外五十多位杰出软件架构师工作经验和观点的精彩提炼。

链接：http://news.csdn.net/a/20100429/218192.html

在麻省理工读计算机专业，看美国的计算机教育

概要：本文通过对比清华和MIT不同的教育方式，映射出中国与北美国家教育相比所存在的差异与不足之处。在中国像清华这样的名校中很难找到一门集理论实践于一体的课程。这或许是中国教育值得关注的地方。

链接：http://news.csdn.net/a/20100206/216928.html

技术总监谈好的程序员如何写代码

概要：本文指出好程序员所写代码整洁而规范，有视觉美感，空白错落有致，注释恰到好处，命名和排版统一而规范，相反差程序员的代码往往含有过长的函数、命名方式和排版混乱、嵌套结构过深、表达式异常复杂。仅观看代码即可判断出程序员的优秀与否。

链接：http://news.csdn.net/a/20100104/216255.html

优秀程序员的45个习惯

概要：2010年初图灵公司出版的《高效程序员的45个习惯：敏捷开发修炼之道》一书在国内热销，而本文是针对该书作的一篇精彩读书笔记。文中摘录了高效程序员的45个习惯，并针对这45个习惯加注读书摘要，提纲挈领地分享了每个习惯的精髓。

链接：http://news.csdn.net/a/20100212/217004.html

2011 IT大趋势

概要：本文指出云计算、社会化（协作）、移动、分析（数据挖掘、商务智能）是值得密切关注和投入的热点。同时列出了Gartner、Forrester和ReadWriteWeb网站总结的值得关注的技术趋势。

链接：http://news.csdn.net/a/20101124/282411.html

Top10 2010年度CSDN十大博客文章

如何做一个出色的程序员

概要：博主石小凯根据自身经历，总结出六条秘籍：爱这行；踏踏实实打好基本功；注重日常积累，厚积薄发；技术上做到既广且钻；培养分析问题能力，善于追根溯源；全面培养能力，不做纯粹“技术人员”；阿里巴巴六脉神剑文化。文章措辞丰富幽默，相信一定可以给开发者带来启示。

链接：http://blog.csdn.net/lkjxshi/archive/2010/01/07/5149541.aspx

面向程序员的数据库访问性能优化法则

概要：数据库访问优化有五大法则：减少数据访问、返回更少的数据、减少交互次数、减少数据库服务器CPU运算及利用更多的资源，每个法则下均列出了详尽的解决方案和案例分析。全文结构清晰，内容翔实，可以作为开发者优化数据库实践的参考资料。博主叶正盛，就职于阿里巴巴，负责数据库应用与开发。

链接：http://blog.csdn.net/yzsind/archive/2010/12/06/6059209.aspx

不要给技术人员做绩效

概要：本文阐述了在非超大型公司中应该取消技术人员的绩效考核。技术人员的绩效很难用表格来考核，而且还会打消技术人员的工作积极性。与其用冰冷的绩效去考核，不如在日常工作中多沟通，塑造良好的文化和工作氛围。博主潘燕辉，资深媒体人，现任网际快车副总裁，负责技术和产品。

链接：http://blog.csdn.net/panyanhui/archive/2010/07/01/5706345.aspx

代码背后的点滴

概要：本文指出语言和平台不是影响当前互联网应用发展的重要因素，代码背后的设计思想才是关键，并举例说明了在11种特定设计场景中所应采取的设计思想。“用最熟悉的方式去表现最自然的想法，那才能做到游刃有余”这一观点得到了网友的大力支持。此文犹如一杯醇香的咖啡，很值得慢慢去品味。博主岑文初，淘宝开放平台主架构师。

链接：http://blog.csdn.net/cenwenchu79/archive/2010/09/09/5872324.aspx

侃侃那些美丽的编程语言

概要：本文分析了Python、Haskell、Smalltalk这三种语言。所有语言都各有千秋，如何选择要根据工作需要。了解语言本身，在不同的环境下选用合适的语言，可以达到事半功倍的效果。作者韩祝鹏，任职于小米科技。

链接：http://blog.csdn.net/albert_lee/archive/2010/05/11/5578781.aspx

Linux编程的108种奇淫巧计之编译展开（系列）

概要：本文是《Linux编程的108种奇淫巧计》系列中的一篇，讲述了第四种技巧：编译展开。此技巧在高质量开源代码中经常使用，可将循环语句展开到任意层次。全文提供了完整代码、编译和运行方法，方便开发人员亲身体验。博主梁斌，《走进搜索引擎》一书作者。

链接：http://blog.csdn.net/pennyliang/archive/2010/10/28/5971059.aspx

五分钟讲清楚UML

概要：本文从产生原因、形成过程、高效性、使用方法、重要性五个角度逐步介绍了UML技术，属于快速入门教程。全篇图文并茂、通俗易懂。博主为蒋彪。

链接：http://blog.csdn.net/nanjingjiangbiao/archive/2010/04/28/5540126.aspx

程序员应知——把小事做好（系列）

概要：本文是《程序员应知》系列中的一篇，告诫了程序员要戒骄戒躁，把小事做好。工作中虽然一些看似简单、重复、技术含量不高的“小事”，但要做好并不容易。做好小事，不仅能考验程序员的细心和耐心，也能赢得他人信任；小事做好，才能担当大事。另外，把小事做好，并不是机械地蛮干，而应善于思考，思考高效率的做事方法，才能提升自己。博主侯伯薇，就职于大连某保险公司。

链接：http://blog.csdn.net/lingyun2005/archive/2010/07/27/5767972.aspx

你不是菜鸟——给一位菜鸟MM的回信（系列）

概要：性别和学历并不是影响步入IT行业的关键因素。一旦踏入IT这个行业，就犹如攀登高山一样，不要上下观望，只需一步一步向上不停攀登，终可达到成功彼岸。博主钟声，著有《Java程序员，上班那点事儿》一书。

链接：http://blog.csdn.net/useway/archive/2010/07/19/5746085.aspx

C++0X的三件好东西

概要：技术学习者在学习新技术时，应着重掌握技术的核心内容，对于次要部分，留到实践中学习即可。以C++0X为例，Rvalue reference、function/bind、Lambda expression和closure是它的核心内容，是学习重点。博主孟岩，现就职于IBM（中国）有限公司企划传播部，曾任《程序员》杂志技术主编。

链接：http://blog.csdn.net/myan/archive/2010/09/11/5877305.aspx

Top 10 2010年度CSDN十大热点资讯

李德毅院士：云计算将颠覆软件产业格局

概要：李院士就云计算产生的大背景，以及与此相关的计算资源虚拟化、云计算标准、云安全三个热点展开深入探讨。李院士认为云计算的未来有两个趋势：第一是在移动互联网上的云计算将来得更快；第二，云计算的推进方式会是从服务到互联网再到物联网。

链接：http://www.programmer.com.cn/3556/

Linux之父Linus Torvalds：C++一无是处

概要：Linus Torvalds认为C++想解决的问题不对路，都是一些皮毛，没有涉及真正深层次的问题。C++的对象、模板和函数重载基本上纯粹是C的语法扩展，反而把C的语法和类型系统弄得更糟。他建议，在系统编程里直接用C就可以了，非系统编程里，应该选择一种有垃圾收集的语言，C++的特性基本无用，只会捣乱。

链接：http://news.csdn.net/a/20100612/218785.html

悼念一个伟大的公司——Sun

概要：2010年1月21日，甲骨文以70亿美元收购Sun的交易获得欧盟批准，这意味着全球第二大商用软件提供商甲骨文正式进入硬件产业。在规模达170亿美元的高端计算机服务器市场，Sun曾是市场的领先者。然而从这天起，这个技术领袖辈出、曾经创造了众多神作的伟大公司，离开了我们。

链接：http://news.csdn.net/a/20100122/216640.html

Google受攻击技术细节曝光

概要：引述McAfee公司的话说，（攻击Google）黑客使用了前所未有的技术，组合了加密、隐秘编程技术和IE中的未知漏洞，意图窃取Google、Adobe和其他大公司的源代码。业内资深专家表示，除国防工业以外，从未见过商业公司遭受过如此复杂程度的攻击。

链接：http://news.csdn.net/a/20100115/216509.html

组图：2010年30个最佳iPhone应用

概要：iPhone应用已经发展成为一个拥有50亿下载量和10亿美元产值的产业，苹果App Store里的应用也已超过20万款。然而，走遍App Store，你会发现大多数软件都已经出现，而且至少被创新过两次。各种复制品琳琅满目，使高质量的应用更难吸引消费者的注意。本文向大家推荐了今年发布的30款最好iPhone应用。

链接：http://mobile.csdn.net/a/20100809/277934.html

程序员能力矩阵，你属于哪一层？

概要：以图表的形式对程序员的能力矩阵进行了分级，从包括计算机科学（数据结构、算法、编程体系）、软件工程、程序设计、经验值、学识力等多角度考察，共分为四个层次，每个层次的知识都是渐增的，即层次n也蕴涵了所有低于层次n的知识。

链接：http://news.csdn.net/a/20100322/217537.html

只有10%程序员能正确实现二分查找算法

概要：作者在贝尔实验室和IBM考察员工的时候，都出过这道考题，那些专业的程序员有几个小时的时间，可以用他们选择的语言把二分查找写出来，高级伪代码也行。考试结束后，几乎所有程序员都认为自己写对了。但是，经过半个钟头测试用例的验证，结果是90%的程序员写的程序中都有Bug。

链接：http://news.csdn.net/a/20100423/218099.html

360 VS QQ：一场精心策划的西安事变

概要：作者将公司比喻成国家，公司之间的竞争也类似战争，国外网站曾经有一幅描述微软帝国的作战形势图就是例证。这次360和QQ之争，在中国软件互联网史上，是一场不折不扣的超级大战。对这场战争的分析是：一场精心策划的西安事变。

链接：http://news.csdn.net/a/20101105/281495.html

IT蚁族：蜗居和逃离

概要：北京蚁族群体在大学里的专业五花八门，从事的工作类型也不尽相同，但大多数都与IT产业有关联，要么身在IT类公司，要么从事IT类的技术工作。到底是什么吸引着蚁族甘心蜗居在IT产业中？蚁族真实的生活状态又是怎样？

链接；http://news.csdn.net/a/20100316/217473.html

江民科技创始人王江民因病去世，享年59岁

概要：他是中国最具特色的程序员创富典型之一，38岁开始学习计算机，两三年之内成为中国最出色的反病毒专家之一，45岁只身一人独闯中关村办公司。这些事实已经让业界无法忽视王江民对中国软件开发带来的影响力。尽管现在KV系列产品中早已没有了一行他的代码，但毋庸置疑，KV系列让众多的程序员知道了王江民。而他身残志坚的毅力和品质也让很多程序员面对困难和挫折时，从中得到鼓舞。

链接：http://news.csdn.net/a/20100405/217737.html



“程序人生”栏目主持人

周筠，电子工业出版社博文视点（武汉）负责人。带领团队策划出版了《编程之美》、《代码大全》、《程序员修炼之道》、《走出软件作坊》、《我是一只 IT小小鸟》等图书。关注技术的同时，更关注技术背后的个人和团队。

十年莽撞IT路

■ 文 / 冯大辉

博文视点周筠老师多次叮嘱我写点关于个人成长的心得，颇感压力。回首在IT这个行业也差不多有10年了，没做成什么惊人的事业，也没怎么赚到大把的钞票，如果冒充什么成功导师大谈人生感悟岂不是会被读者朋友笑掉大牙，当然引来板砖也说不定。如果说有可取的，恐怕也就是自己莽撞地混入这个行业，阴差阳错地一路走过来，有些教训或许能供朋友们参考一下。所以，硬着头皮记录一下过去几年的流水账。是为序。

程序员

每当被《程序员》杂志约稿的时候，我都要提醒一下自己：我不是程序员。这是真心话，我没有做过一天真正意义上的程序员，尽管我非常想做。我不太喜欢程序员自嘲称自己为“IT民工”或是“码农”什么的，总觉得做这个行业，就要尊重自己的职业才是。我自己并非计算机科班出身，在大学里也没能积累下足够多的写代码的经验，毕业求职的时候其实是没办法竞争编程开发相关的岗位的，所以只能走差异化竞争路线。幸运的是，我认识到IT行业中除了开发程序其实还有其他工作岗位可以选择。所以，我较早地逆向推演出自己能够做哪些事情，并且结合自己的兴趣，在操作系统（Unix）的实践和网络（比如TCP/IP）理论方面下了一番苦工夫，加上一些机缘巧合，最后能够有幸撞入这个行当。



丁香园网站CTO冯大辉

在2000年左右，专业歧视还是蛮严重的，几乎绝大多数IT公司的校园招聘都会比较严格地限定专业，而像我这样从八竿子打不着的生物学要跨入IT行业的，基本上很少有人理睬。我想我永远都会感谢给我机会的那位面试官。

招聘我的公司是个中字头的国企，有着较为荣耀的历史，在当时来看，整体上是一家大公司，但具体到每一家分公司，则是不折不扣

的小公司，当然也谈不上什么好的公司文化。因为毕业之前没有和公司联系，所以也没有到公司实习，毕业后就直接到北京来报到了。行政人员还为是否留下我这个人作了一番斟酌，想来也挺有趣。我们这几个毕业生的岗位是系统工程师，说得直接一些，其实也就是做一些Unix下软件的安装和实施，需要到各地出差。这对刚走出校园、毕业前甚至没有出过东三省的我来说也是一种很好的锻炼，可以了解一下各地的风土人情。这份工作也磨炼了我与人打交道的能力，尽管做得还不够好——还不能很好地控制自己的脾气，暗地里也因此吃过不少亏。

没有项目的时候，有大量的时间与几个同事一起学习和实践。就是在这家公司，我在Unix操作系统之外，开始选择数据库作为一个学习方向。没有人告诉我应该做什么样的选择，说老实话，只是看当时招聘数据库管理员的公司开出的薪水的确都很不错，就误打误撞开始了数年的数据库技术之路。国内当时出现了一个面向数据库技术的网络论坛——ITPub.net，聚集了一大批数据库技术的爱好者，大家在论坛上分享资料，交流心得，不亦乐乎，也结识了不少朋友。正是互联网给了我们学习更多知识的可能，否则，只有在具体的应用场景才有可能接触到这些，而我也走不到今天。

当社交面逐渐打开的时候，你接触的信息也会越来越多，所谓的机会，其实是相当多的。

在这家公司工作了一年多，感觉自己的数据库有了一点基础，能力有了提高，就冒着极大的“风险”跳槽了。说是风险，因为母公司隶属国企，每个毕业生都签订了四年的合同，如果提前离职要对公司进行赔偿，合同上写明总计两万五千元，对那时的我来说，这是一笔巨款。当然现在想可能也没什么。我当时倍感压力，有点杨白劳遇上黄世仁的感觉。我想现在的毕业生应该很少再面对这种霸王条款了吧。有朋友会说，毕业一年就跳槽，太对不起这家公司了吧？其实我也是不得已而为之。

数据库

很多朋友知道我，是因为我曾经在DBA这个岗位工作过很久。实际上，我走向DBA这个岗位并非一帆风顺，甚至稍有一些周折。我新入职的这家公司隶属于一家更大的国企，公司负责人雄心勃勃，组建了一支不小的数据库技术团队，主攻电信行业商业智能市场。遗憾的是，最后在商务上并不理想，而且项目实施也出现了不小的问题。当然，那是在我离开之后的事情了。我在这个团队只工作了半年左右，并不顺利，也不开心。问题主要是出在我自己身上，和直接主管的沟通总是有问题。遗憾的是，我当时甚至不认为自己有问题，这是很多职场新人的通病，或许也是很多人的通病——总喜欢把责任推到别人身上，而无知地认为自己没有错。我在这家公司的一个收获是看到了余世维的一个讲座视频，应该是给某电信企业做的培训课程吧。虽然现在看来，这套成功学的东西其实没什么可取之处，但当时给我的感觉还是挺震撼的。因为此前，我从来没有考虑过如何修正自身的一些问题。

从这家公司离开后，在接下来的这家公司的经历则颇有戏剧性。我作为数据库管理员入职后没多久正好赶上非典爆发，第一次享受在家远程办公的乐趣。非典过后上班第一天得知，老板居然把公司卖掉了，就这样，没做什么事情，拿了几个月工资，但并没觉得自己赚到了什么便宜。考虑到并入新公司后发展方向对自己并不有利，所以我不得不考虑再次换一份工作。其实这个时候对我自己来说，已经有些苦恼了，我知道频繁地更换工作对一个人发展的负面影响是很大的，尤其是在刚进入这个行业不到两年的情况下。

我下决心，无论如何下一个工作要做更为长久一些。

很快，我又找到了一家公司。新公司规模不大，但总算也是一家外企，老板是韩国人，有美国留学的背景。能够进入外企可能是那个时候很多人的一个阶段性目标，怎么说呢，至少我在当时还是很羡慕一些在外企工作的朋友，起码薪水很不错。谁有那么长远的眼光不在乎薪水呢？甚至也很少有过来人跟你讲关于发展与职业规划这些事情，基本上是凭着感觉走，只能多观察、多分析别人的经验。新公司的同事都很有活力，大家关系相处得也都不错。公司制度比较灵活，我甚至有时候中午才



冯大辉@798

去上班。当然，公司给出了空间，工作起来也都是挺卖力的。

这家公司有两个业务方向，一个是给联通做增值服务，我的工作职责包括在这一部分中；另一个是开发手机上的浏览器，这是公司发展的重点，大部分同事也都是做开发的。现在回想起来，这个浏览器太超前了，当时是2004年左右，手机根本没有发展到这个地步，而且，单靠这一款软件，没有上下游的产品支撑，尽管有想象力，但最后还是没有合适的出路。一年多之后，公司启动了另一个产品——在线音乐网站。这也是老板借鉴韩国的互联网模式做出的决定。很显然，结局不猜也会知道，同样是个失败的项目。每每想起这个项目我甚至有些后悔，错失了一个很好的锻炼机会，如果当初能够多承担一点点责任的话，或许能做得更好。

我是在加入这家公司之初开始更多地关注起互联网技术，搭建了自己的个人站点，后来尝试写起了Blog，通过捣鼓（真的是捣鼓）个人站点，一点点地摸索、学习到了更多的东西。我对Web相关的一些技术没有系统的学习过，只是时间长了形成了感觉而已。早期Blog技术圈都是一些很纯粹的技术爱好者，更多的人只是为了分享和乐趣而写作。通过这个途径，我慢慢结识了另外一个技术群体。做我们这个行当的技术人员总会抱怨没什么前途，没什么空间，根据我的观察，社交面太小也是对很多人的一个制约。当你社交面逐渐打开的时候，你接触的信息也会越来越多，所谓的机会，其实是相当多的。

杭州五年

2004年，一位素未谋面、但是在社区内打过不少交道的朋友邀我加入阿里巴巴。刚刚启动的支付宝数据库没有人维护，服务器压力也不大，这个工作非常吸引我。尽管上一年也有类似的机会，但当时感觉一是自己技术未必能撑起来，二是薪酬似乎也很一般，再者阿里巴巴当时的声誉并非很好（竞争对手散布了很多妖魔化阿里巴巴的信息）。经历了这一年，面对邀请，我忽然觉得它可能是个机会，毕竟再一再二不能再三啊，万一错过了呢？去看看也不会损失什么，而且阿里数据库团队已经拥有了好几位技术社区的牛人，能吸引这么多人才，也说明公司肯定有自己的特点。

杭州一行，接触下来感觉这帮家伙都太有趣了，我坚定了自己“南下”的决心。马云在2004年底发表的“2005年将是中国电子商务安全支付年”的言论，让我很受触动，想想做的事情有可能给互联网带来一些改变，那将是让人多么欣喜的事情，一时间我对杭州的这份新工作充满憧憬。另外，当时我对浙江的商业气氛也很感兴趣，“在那边工作几年，学习一下浙江人怎么做生意，然后再回北京”，我用这个说辞说服了女朋友，我现在的太太。

到了杭州之后工作就上了快车道。支付宝

当时正面临着一次相当大的业务改造，为了避免对用户的影响，很多操作都要夜里进行，白天还要支持开发团队。前三个月的工作强度之大让我始料未及，但又不能临阵逃脱，只能硬扛。到了项目发布前夕，连续奋战几个通宵，人都熬不住了。正式发布那天遇到了大麻烦，悲剧来临，因为之前赶进度而忽视了性能方面的问题，导致发布时性能问题恰恰成了拦路虎。此时整个技术团队都在后面看着你，真让人一筹莫展，这种情况下要承受的压力可想而知。让我至今感激的是，团队里的其他几位同事在我撑不住的时候顶了上来，直到最后系统上线，大家终于松了一口气。我不知道有多少人在工作中面临过类似的压力，那段时间，每



冯大辉曾受邀主持SD2.0大会

天早晨起床后，我告诉自己，坚持过今天就好了……就这样，一点点熬了过来。在以后的几年，我甚至遇到了更大的压力，但因为有前面的铺垫，抗压能力已经加强了许多，甚至于我很享受这个过程。一旦进入IT这个行业，早晚都会遇到你职业生涯中的种种看似跃不过去的障碍，我所能给出的建议也无非就是“再坚持一下，不知不觉就跨过去了”。

这段时间我甚至强化了另外一个习惯——阅读。每当压力巨大的时候，为了不失眠，需要切换一下思维，临睡前就强迫自己看一会儿小说，效果还不错。我不知道自己什么时候练成了快速阅读的能力，应该是长期积累的结果吧。小时候接触不到太多的可以阅读的东西，所以有些书籍甚至会读个几十遍，最多的一本书恐怕足有上百遍吧。到了读大学的时候，有了更大的阅读空间，我成了同学中到图书馆借书最勤快的人，以至于图书馆管理员都认识我了。当然我还省吃俭用买了更多的书，包括学习一些新技术的时候。我还有一个习惯就是同时会多买几本书对照着阅读，就我个人而言，收效还是不错的。我比较喜欢有阅读习惯的技术人，当面试的时候，如果一个技术人很长时间都没有读过一本书，会让我降低对他的评估分数，当然，这是我的个人偏见而已。应该说，对于我而言，阅读是一种乐趣，和有些人喜欢电子游戏是一样的。

在杭州的前三年，为工作牺牲了自己的不少业余时间（正常的工作时间有时候反而比较空闲，所以，才有可能写一些文章），因为公司随时可能有事情，而且有事情就不会是小事情，大多数业务都直接涉及到资金数据，稍有不慎，可能就会酿成大祸。我现在非常怀念和同事们通宵发布的那些日子，的确非常辛苦，但其中也有莫大的乐趣。每当即将拂晓的时候，在崔健的音乐声中，看着窗外渐渐清晰的景色，总让人有一种莫名的欣喜，好像我们每个人的前途都光明起来。

公司的业务实在发展得太快，技术要想不拖业务的后腿，也只有跑得更快。强迫自己做更多的事情，这也是那几年我在技术上有点进步的主要原因。慢慢地，我意识到，要更好地提供后端的数据支撑能力，不理解整体的技术架构是不行的，这是我开始学习Web架构方面知识的起因。翻看我自己站点早期的关于网站架构的文章，其实没什么技术含量，无非是一些分析各个网站架构的笔记而已。最开始记录的时候我只想写给自己看。有些东西，看过了不一定理解，理解了不一定能写出来，写出来但不一定能说明白，说明白不一定让别人也明白。我只是从初级阶段做起，把看过的东西做个记录，然后通过读者的反馈再做一定的梳理，有机会的话尝试给别人讲讲，把这当成锻炼自己的一个途径。让我始料不及的是，这类文章受到了很多读者的关注，让我倍受鼓舞，于是写数据库相关的内容越来越少，写Web相关技术的文章越来越多。不知不觉，我又完成了一次技术背景的转换。

2008年上半年，由于自傲和莽撞，我再次犯了一次意识上的错误，而被动地导致了一

次工作职责上的调整，或者说被放逐了一段时间，临时成了一个“架构师”，也变得相对清闲起来。于是我也有机会进行一些面向外部的技术交流，可以说打开了另一个天地。通过这些交流，让公司一些真正的大牛为业界所知，引进了一些技术人才，侧面改善了公司在技术社区的形象，这是让我很欣慰的一件事儿。当然，这些招摇过市的事情也会引起一些误解，但是没有人做事情能面面俱到，不是吗？

过了一年多，我又被召回数据库团队。作为团队的管理者，在新的岗位上有很多东西需要学，也颇有挑战，但我也越来越觉得自己想要做的不是这些事情，再过几年，也无非重复一些以前的事情，将一些业务数据做得更高、支撑能力更强而已。在这个前后，我三十而立了，我结婚了，我的人生观和价值观不可能不发生一些变化。我更想看到自己的一些想法变成现实，我喜欢通过努力让产品尽快地改进，我热爱互联网而不喜欢金融的刻板，我喜欢社区，喜欢开源文化，喜欢Twitter……不过我讨厌终日繁复的会议……朋友们，你们中一定有人听过“家猪”和“野猪”的故事，我发现自己不知什么时候已经成为了一只“野猪”，再也不能变回“家猪”了。

支付宝现在已经发展成为了一家不折不扣的大公司。我觉得自己是个幸运的人，亲身经历了一家公司从初创到发展壮大的过程。作为一个以技术安身立命的人，在这个过程中我观察到、学习到的东西比什么都重要，我暗自庆幸没有一味低头干活，有时候也抬头看了一下路，这是我真正收获到的。在阿里巴巴历经五年多，我深刻感受到阿里巴巴是一家了不起的企业，有独特的魅力，将来也定然会发展得更好。更令人怀念的是这里有很多优秀的同事，我从他们身上学到很多，和他们一起战斗的日子让我永生难忘。生命中非常重要的五年留在了这里，要感谢的实在太多。

丁香园

写这篇文章的时候，我已经在丁香园（DXY.com）工作将近半年了，我很享受创业的状态。在做这个决定之前，我曾经问过自己，你有足够的管理经验吗？非常遗憾，没有。不过，我还可以继续学习，还可以不断地改进自己，能够帮助团队迅速成长，我的头脑还没有完全僵化，还能够“坚持”下去，所以，我有信心接受一次新的挑战。

如同当初加入支付宝的理想主义一样，我想通过丁香园这个项目，在医疗健康领域，能够帮助一些人，让这个环境稍微美好一点。做这些事情，不一定会让自己功成名就，但是有可能让自己心里更加安宁一点、更加踏实一点。不是每个人到这个世界上最后都会富有，但如果能够健康生活，能更加快乐一些，这比什么都重要。

后记

我是不是遗漏了什么？回头看这篇文章，我发现仍然少写了很多或许关键的内容，也似乎潜意识中将这个过程美化了许多，选择性地遗忘了许多，所以写得轻松了许多。在杭州这几年，因为不适应气候等诸多因素，每年都有几次很严重的关节疼痛发作，痛不欲生。你看，这只是我这十年付出的代价之一，类似的苦楚还有更多。有时候想，如果我当初不来杭州会怎样？或者说如果当初不选择IT会怎样？过去这十年中，我做过不少次选择，根本不知道如果做另一种选择的话，我现在会是什么样子。大家应该知道电影《黑客帝国》的那一幕，选择红色药丸还是蓝色药丸，所面对的世界将截然不同。

有时候被即将毕业的同学问到关于职业或人生方面的问题，我真的不知道该如何作答，不知道每一次选择会怎样改变你，因为每一个个体是不相同的，不可能复制别人的道路。但有一点可以肯定的是，在这个糟糕的时代，我们都将面对更大的压力，历经更多的痛苦，唯有更加顽强一些，在你快绝望的时候再坚持一下。这是我经历十年莽撞IT路后给自己的一点忠告。

作者简介 | 冯大辉



现任丁香园（http://www.dxy.cn）网站CTO。曾历任支付宝架构师、数据库团队负责人等职。

■ 责任编辑：董世晓（dongsx@csdn.net）

这个圈圈不简单

■ 文 / 潘加宇

用例背后隐含的是市场的思维，对仍习惯于学生思维的开发人员来说，用例是一种折磨。

用例提出“卖”字

用例就是：给定一个研究对象，它的执行者乐意“买”，该研究对象又能“卖”的一个价值。

还是以被说滥了的取款机为例：储户来到取款机面前，插卡输密码，拿到两百元后，到隔壁投注站买了50张即开彩票，哇噻，中奖五万元。那么，取款机的用例到底是什么，“登录”、“取款”还是“中大奖”？或者说“都对，从辩证的角度看嘛”？

“登录”不是取款机的用例，因为取款机不能这么叫卖：“来啊来啊，我这里能登录啊”，然后储户就说：“哇，真棒，这正是我想要取款机提供的服务。”储户对取款机的期望可不止这么多。“中大奖”也不是取款机的用例，储户倒是很想从取款机那里得到这个服务，可惜，取款机没法承诺。只有“取款”是取款机能承诺的（后置条件：系统把钱吐出来），而且储户乐意因此使用取款机的服务。



图1 取款机的用例

用例之前的许多需求方法学，把需求定义为思考系统“做”什么，用例把需求提升到思考“卖什么”的高度。这种思考是非常艰难的，因为它没有标准答案，只有最佳答案。对于仍然习惯于用学生思维思考、喜欢背标准答案的开发人员来说，思考“什么是系统的价值”有时甚至会让他痛苦到想要逃避。

但逃是逃不开的，生活中处处都需要这样的思考。人们求职、求偶、求××……不就是要搞清楚“我”这个人肉系统到底应该卖给谁、怎么卖的最佳答案吗？我想卖给你你不要，想买我的不知道在哪里，就是人生的痛苦所在。许多时候，我们像没头苍蝇一样投简历、相亲，却不舍得抽出时间静下来思考“价值”的问题。

我们来看“程序员”这个人肉系统，它为老板提供的用例是什么：装Visual Studio？编码？为公司赚钱？答案是编码，这是老板对程序员的期望以及程序员可以提供的承诺的平衡点，或者说，这是程序员能卖、老板愿意买的价值。程序员不能因为装了个Visual Studio就理直气壮地向老板要报酬，老板不给就生气；老板也不能因为程序员明明编了代码，由于市场

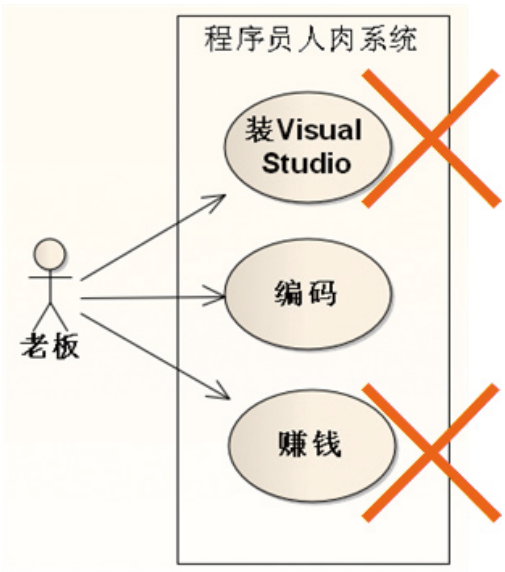


图2 期望和承诺的平衡

不景气软件卖不出去赚不到钱而责怪程序员。

程序员如果摆错了自己的位置，没有好好完成本职工作，反倒是对公司的发展方向大放厥词、指手画脚，老板是不会喜欢的，因为他不期望从程序员身上“购买”这个服务。可见，搞清楚自己的“用例”，认清自己的定位，对人生多么重要。当然，随着时间的发展，这个人肉系统可以不断升级，提供更多的用例。

不断思考用例的过程，就是不断发现价值的过程。如果你通过用例思维不断训练需求能力，可能你出去创业成功的概率就很大。看，用例思维是一个可以让你终身受益的东西。

业务建模帮助理清“价值”

下面我们来研究一个人事系统，员工先通过系统交请假单，主管使用系统批假，然后人事部专员使用系统备案。用例图如图3所示。

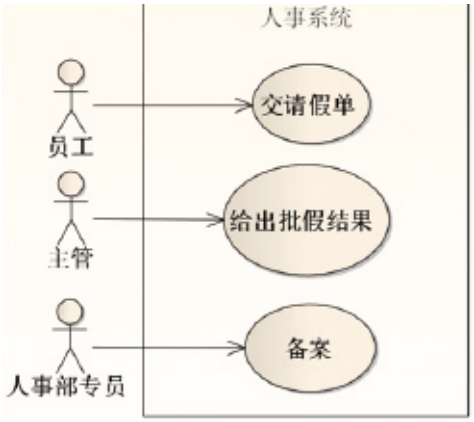


图3　人事系统的用例，版本一：三个用例

但有的时候，开发人员就会想，哎呀，没请到假，怎么能算价值呢，用例图应该是如图4所示才对。

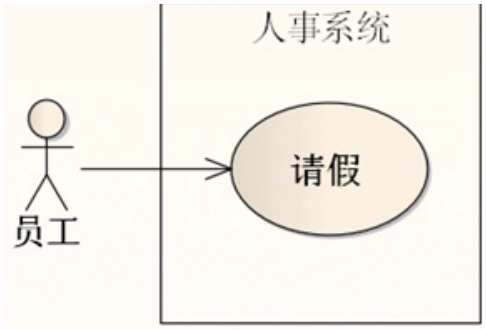


图4　人事系统的用例，版本二：一个用例

这个时候，业务序列图（图5）可以帮我们理清楚系统责任。

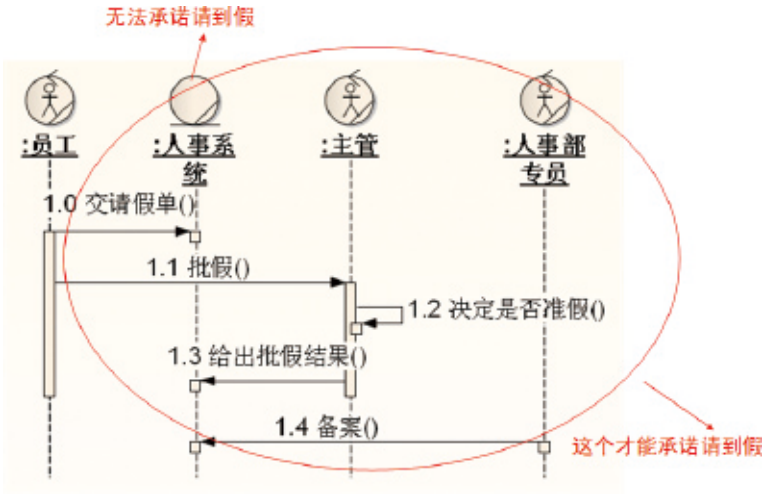


图5　从业务序列图看清系统责任

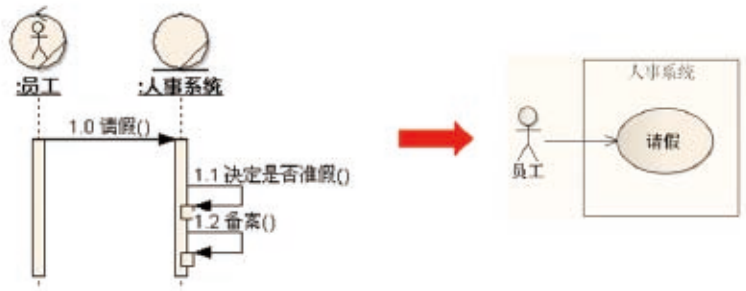


图6　智能批假的业务序列图，请假变成了人事系统的用例

从图5可以看出，人事系统不能向员工提供能请到假的承诺，需要“人事系统＋主管＋人事部专员”打包在一起才可以。如果像图6这样，结果就不同了。

或者虽然没有智能批假，但把研究对象变成一个组织——“请假部”，结果也不一样，如图7所示。

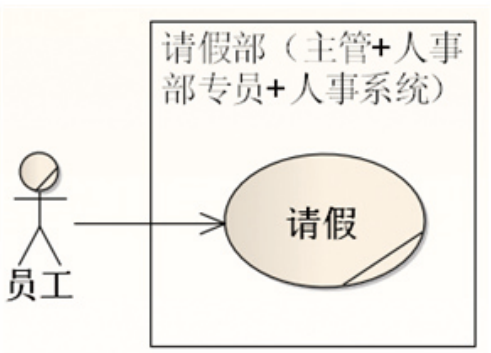


图7 研究对象改变，用例也随之而变

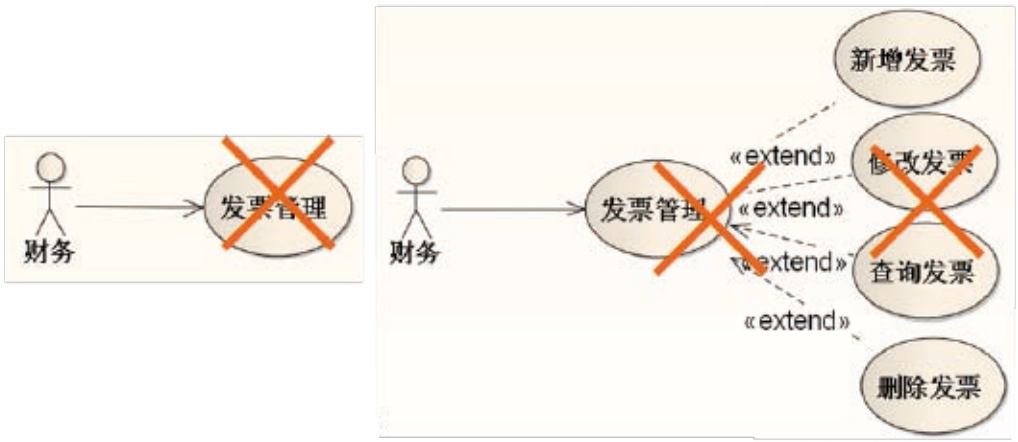


图8 数据库思维导致的××管理

数据库思维不可取

打开一些开发人员画的用例图，看到最多的就是CRUD用例，把数据库的各个表名加上CRUD就得到：新增××、修改××、删除××、查询××。很多用例书籍和文章也提到了这个典型的错误，所以开发人员也学会了，干脆把它们改名叫管理××（或××管理）——可惜，这是换汤不换药。如图8所示。

新增、修改、删除、查询、管理、改变状态……这些词是数据库（或面向对象）里的“鸟语”，不是领域里的“人话”。业务流程中不会有人说，小张等一下，我到系统那里去做一下发票管理，只会说，我去开一张发票、我去作废一张发票、我去开一张红字发票……而且，这些事情以不同的频率发生在不同的业务流程中。所以，正确的用例应如图9所示。

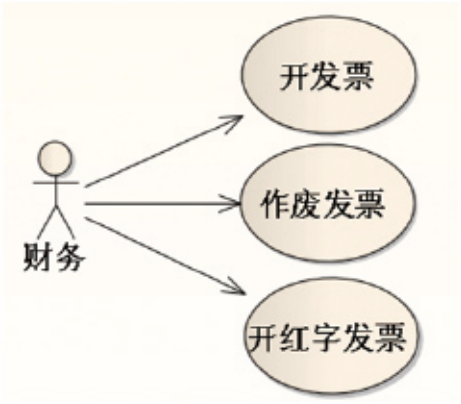


图9 “说人话”的正确用例

通过数据库的角度来定义系统的价值，在现存的信息系统中随处可见。我甚至见到过这样的信息系统：在界面上把所有的“新增”功能都放在一起，根本不管这些功能是给不同的人在不同时间和不同业务流程中使用的。开发人员肯定想“反正都是数据库里新增一条记录嘛”，这样做需求倒省事。可惜的是，这样做除了可以帮助开发人员自我安慰之外，对做出好卖的软件没有帮助，而“让软件更加好卖”正是做需求的目的。

如何慢慢学会“说人话”呢？老老实实去研究业务流程，做好业务建模，尽量从业务序列图中映射出系统用例，这样得到的系统用例是不会骗人的。实在映射不出来的，最后才用“管理××”来打扫战场。“管理××”用例往往是给系统管理员管理基本数据用的，而且都是千篇一律，不必花太多心思在上面。

用例不怕多

上面提到的数据库思维实际上就是开发人员喜欢寻找共性来“复用”用例的一个特

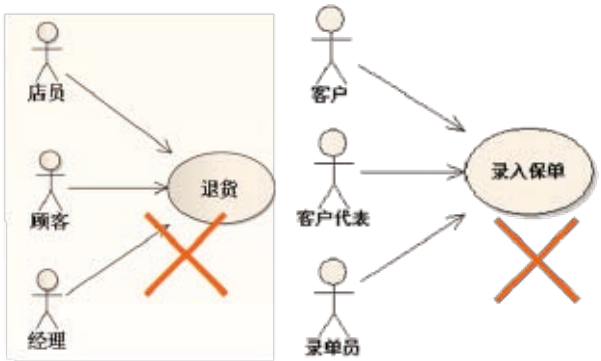


图10　各种“复用”用例的错误

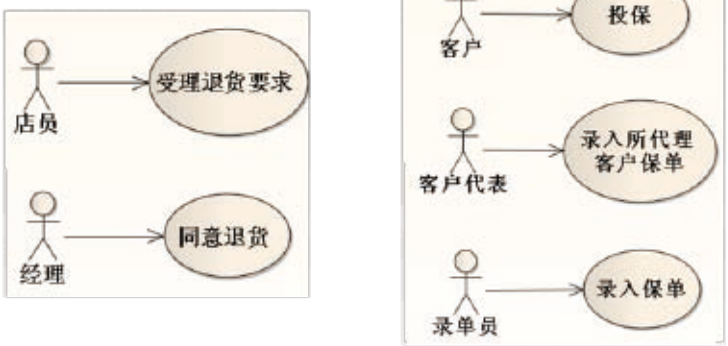


图11　为不同执行者提供不同的用例

例。需求是不考虑“复用”的，考虑“复用”说明已经进入了设计的思维。类似的错例还有：因为都参与了退货的流程，干脆共用一个用例“退货”；因为最终结果都是导致数据库的“保单”表里增加一行，干脆共用一个用例“录入保单”……如图10所示。正确的用例图如图11所示。

退货例子中的错误还比较好避免，保单例子中的错误却是非常容易犯的。从“卖”的角度来看，这是系统同时存在的三种不同用法，涉及到的涉众利益也不同。客户在家里通过网络投保，操心的是“可别上当”；客户代表录入自己代理的客户的保单时，操心的是“佣金要高”；录单员面对堆积如山的待录入保单，操心的是“省力一点”。这些不同，能用“都是往数据库的保单表里插入一条记录啊”这样的理由合并吗?

开发人员不自觉地合并用例，就相当于告诉客户说，你真笨，你买我这些功能，其实我只用了同样几个类灵活组装出来。干脆，你成本价把我的零件买走，自己去组装吧!

说到这里，可能有人就会说了：哇，这样用例岂不是很多？如果说真的按照“卖”的思路去找，确实是这么多，那是好事还是坏事？好事来着！说明你用少数的材料（成本）组装出了很多可以“卖”的价值。事实上，往往系统的用例大量膨胀根本不是因为这个，而是由于开发人员把很多不是用例的步骤当成用例画出来了!

用例思维看我们自己

“人”是一个很奇妙的系统，充分体现了我在本文和之前的文章中不断强调的需求和设计的区别。在外面看，“人”会跑步、跳高、投掷、游泳，但是设计“人”这个系统时，并不会从需求直接映射到设计，得到“跑步模块”、“跳高模块”、“投掷模块”、“游泳模块”。人的内部零件是心肝脾肺肾和五官四肢。同样，也不能从设计推导出需求——因为人有心肝脾肺肾，所以“人”的用例是“心管理”、“肝管理”。跑步跑得好，在现代可以参加奥运会获得大利，再不济也可以送快递谋生，在石器时代至少也可以逃命，“心管理”能干什么呢?

作者简介 | 潘加宇



UMLChina首席专家。1999年创建UMLChina，潜心研究需求和设计技能。2002年开始对外提供UML需求和设计的技术指导和训练服务，客户遍及各个领域的领袖企业。

■ 责任编辑：董世晓（dongsx@csdn.net）



主持人

冯大辉，现任丁香园（http://www.dxy.cn）网站CTO。曾历任支付宝架构师、数据库团队负责人等职。

架构师接龙

许式伟 VS. 张宴



提问嘉宾

许式伟

许式伟，前金山技术总监，WPS Office 2005首席架构师，开源爱好者。目前任职于盛大创新院，致力于改善国内的创业环境，感兴趣的话题包括分布式存储、网络操作系统、Erlang风格编程与最佳服务端编程实践、搜索引擎等。



回答嘉宾

张宴

张宴，金山游戏运营技术中心技术经理。曾任金山游戏官网逍遥网架构师、新浪播客系统工程师。在Web、数据库、视频、负载均衡、搜索引擎、数据挖掘、网游运营等领域有丰富的架构设计、项目管理、程序开发和系统运维经验。

许式伟： 作为系统架构师，您一般会从哪些方面来保证网站的高可用性（降低故障时间）？

张宴： 很多因素都会导致网站发生故障，从而影响网站的高可用性，比如服务器硬件故障、软件系统故障、IDC机房故障、程序上线前测试未发现的Bug、遭受分布式攻击、突发访问人数剧增等。

一套良好的网站系统架构，应该尽可能地避免只有一台服务器、一个数据库、一套软件节点等单点故障的存在。单点故障一旦发生，将直接导致网站服务不可用，恢复正常服务所需的时间也比较长，甚至还可能无法恢复。负载均衡集群、双节点热备、分布式处理等都可以用来解决单点故障，比如提供相同业务的Web服务器、MySQL数据库从库，都可以构建负载均衡集群。一旦集群中的一台服务器、一个服务出现故障，自动实时摘除，对用户来说是不可感知的，不会影响到整个网站的访问，可以为运维工程师留下足够的时间去排查和解决故障。

对于重要的MySQL数据库主库，我们习惯于从硬件层和软件层来实现热备，避免单点。越是复杂的设备，发生故障的概率越大。在磁盘没有损坏的情况下，应用程序导致服务器宕机的概率，远高于简单的磁盘阵列宕机的概率。所以，从硬件层解决的话，可以在两台服务器上安装相同的数据库版本、进行相同的配置，用SAS或SCSI线连接一台磁盘阵列，将数据库数据文件存放到盘阵上。正常情况下用服务器A挂载盘阵分区，启动MySQL，绑定虚拟IP；如果服务器A宕机，则用服务器B挂载盘阵分区，启动MySQL，接管虚拟IP。从软件层解决的话，则可以借助DRBD等软件做镜像。

IDC机房发生故障的概率较小，但如果发生的话，影响面也是最大的。如果所有服务器都托管在一个IDC机房，一旦该机房遭遇长时间流量攻击、断电、断网、地方政策性封网等，通常只能联系IDC去处理，除此之外束手无策，解决时间也比较长。如果成本允许，将网站服务器分布在两个以上的IDC机房，当某个IDC发生故障时，可以临时切换DNS域名解析来优先恢复服务。

虽然程序代码上线前，经过了测试人员的严格测试，但测试环境和生产环境毕竟有差异，所以一些会急剧影响性能、正常服务的Bug往往在程序上线之后，才会被发现，这就要求我们在发现Bug后，能够迅速回滚到上一正常版本。我们在SVN的基础上，开发了Web代码发布系统，会将每个发布版本之间的文件变更记录下来，一键实现程序代码在多台Web服务器上的发布和回滚。

遭遇DDOS分布式拒绝服务攻击，使用防火墙来对付半连接、假IP，还算比较容易。而那种专挑复杂动态应用程序URL进行的分布式CC攻击，来源为真实IP、真实HTTP请求，具有模拟正规浏览器User-Agent、单个IP的每秒请求数不高、有成千上万个攻击源等特征，很难与正常访问区分开，比较难对付。但是，正常通过浏览器访问一个URL，会加载该URL中引入的JavaScript脚本、CSS样式、图片等文件。遇到CC攻击，需要及时分析日志，找出访问量异常上涨的URL，然后用事先写好的shell脚本找出哪些IP的请求只访问了该URL，而不加载该URL引入的文件，对这些IP进行自动封锁。

系统架构设计时，需要事先考虑到高于目前访问量多少倍的突发访问。对于网游站点来说，访问量受广告集中时间段投放、线上活动的影响较大，带宽峰值时间不固定，对于静态内容，可以使用商业CDN，按实际使用量计费。对于动态内容，如果遇到突发访问人数剧增，超过现有服务器处理能力，最简单的临时处理办法就是增加服务器。上架新服务器需要时间，但是，同一个IDC机房内，可以借助其他业务的服务器，在不同端口开启一组新进程，加入到原有负载均衡池中。另外，可以临时关闭一些Web中的次要功能，来减少服务器消耗。

许式伟： 您在任务切分上，有什么经验分享？您通过哪些手段保证任务的独立性？

张宴： 相信很多人都遇到过这种情况：在一个老项目上修改、增加一些新功能所花费的时间，不比重新来做一个包含所有功能的新项目时间用得少。一个需要长期维护的项目，不可避免地会面临老员工的离职、新员工的接手，很多时候，项目代码的可维护性将决定一个项目的生存周期。让一个新员工在规定开发时间的压力下，去面对一个文档不够详细、陌生的、功能复杂的庞大项目，短时间弄明白所有功能逻辑不是一件容易的事。所以，任务需要切分，将一个大的任务切分成一个个小模块之后，各模块之间可以做到代码独立，互不影响，可维护性也大大增强。

关于任务切分，我以本人今年负责的两个重要项目架构设计为例来介绍一下。在第一个项目：金山游戏官网的《用户行为分析系统》中，由于数据挖掘计算需要消耗较高的内存、CPU资源，一台服务器的处理能力不够，而商业的分布式数据仓库价格又太贵，所以，只有从程序应用中下手，进行任务切分。我们先按需要挖掘的数据指标，将整个数据挖掘任务切分成多个数据挖掘插件，每个插件可以在不同的服务器上运行，多个插件可以同时在多台服务器上。多个数据挖掘插件之间，如果用到相同的某项数据，那么，就将该项数据以冗余方式，复制几份提供给需要的插件，从而实现插件之间无交互、无关联，保证了超大数据量下插件的运算速度。

在第二个项目：金山游戏新版运营管理系统中，则将整个任务切分成了PHP Web管理界面、PHP Web API功能接口、C/C++中间件引擎三部分。这是一种分层结构切分，最上层的“PHP Web管理界面”调用“PHP Web API功能接口”，“PHP Web API功能接口”调用运行在游戏服务器端的“C/C++中间件引擎”，“C/C++中间件引擎”与“游戏服务器端进程”通过TCP、UDP二进制协议、信号、命令行等多种方式通信。四者之间相对独立，代码无关联，通过一层层API接口实现交互。“PHP Web管理界面”负责通用界面实现。“PHP Web API功能接口”内部，又按接入的游戏模块、子功能模块进行了更细的切分，各功能模块之间通过内部API交互。“C/C++中间件引擎”大而全，不处理具体指令，但兼容TCP、UDP、HTTP、HTTPS/SSL、信号、命令行等大多数通信方式，负责和各种类型的游戏服务端交互。这是一套完全由API接口驱动的系统架构，一款新游戏接入运营管理系统时，只需在“PHP Web API功能接口”中增加一个模块；一个游戏新管理功能的增加，只需要在“PHP Web API功能接口”中增加一个子模块。通过任务切分，将复杂功能简单化，也将原来接入一款新游戏所需要的几个月时间，缩短为1~2周。

许式伟： 您通过哪些手段，来保障产品的质量？您倾向于多久更新一次您的网站？

张宴： Web产品质量主要体现在架构、功能、性能、安全、代码唯一性、兼容性等方面。

架构方面，我会先设计一套架构方案，然后让和项目相关的人员、专家组成员参与进来，一起探讨和论证架构的利弊，提出改进意见，保证架构的可行性。所有重要项目的技术方案需要经过专家组的评估。

功能、性能方面，则会由专门的测试人员进行功能测试、压力测试、安全扫描，测试环境分为线下测试环境、线上准测试环境。

在代码唯一性方面，我们开发了一个Web配置信息管理平台及相关PHP扩展，提供给系统工程师，用于配置信息的统一管理。在新项目中，PHP程序配置文件中将不再出现MySQL、Memcached等各类IP和端口信息，统一用Web配置信息管理平台给出的变量代替。从“开发环境→线下测试环境→线上测试环境→线上正式环境”，连接的数据库各不相同，导致PHP开发工程师经常搞混淆或忘了修改，通过Web配置信息管理平台，使得PHP代码中的配置文件，在四个环境中无须作任何修改，保证了代码的一致性，降低了出错率，从而确保了产品质量。

在兼容性方面，我们从操作系统到PHP、MySQL版本，都保持开发环境、测试环境、线上环境的统一，所有的Web服务运行在CentOS Linux系统上。由于大多数PHP程序员习惯于在Windows上编写代码，而我们的程序中调用的一些接口、PHP扩展，只能在Linux下运行。为此，我们开发了一个小工具，可以将多名程序员在各自本机Windows上搭建的nginx虚拟主机、编写的程序文件，映射到一台Linux服务器，用Linux上的php-cgi执行Windows上的PHP代码。这样，PHP程序员修改完本机代码，保存一下，即可调试，多人之间互不影响。自己调试通过后，可以在Windows直接点击鼠标右键，将修改的代码提交到SVN版本库。

Web 2.0时代，讲究网站更新的实时性，动态网站不用说，静态网站的内容发布也要保证实时。我们开发了一款名为Sersync的开源软件（http://code.google.com/p/sersync/），使用Linux 2.6内核的inotify监控Linux文件系统事件，被监听目录下如果有文件发生修改，Sersync将通过内核自动捕获到事件，并将该文件利用rsync同步到CDN源站服务器。Sersync仅仅同步发生增、删、改事件的单个文件或目录，不像rsync镜像同步那样需要比对双方服务器整个目录下数千万的文件，并且支持多线程同步，因此效率非常高。金山游戏官网的CMS内容发布系统，无论网站编辑通过Web还是FTP上传图片、视频、附件，还是系统工程师直接去CMS发布服务器上增加、修改、删除文件，干完这些事情后不用做任何处理，Sersync 会自动将发生增、删、改事件的文件同步到CDN源站服务器，并可以在文件同步完成后，自动调用CDN缓存刷新接口，主动刷新发生修改、删除的文件的访问URL。

许式伟： 您在面试时，通常关注应聘者的哪些方面？哪些问题经常会问呢？

张宴： 第一，需要具备岗位要求的基础技能知识，这方面我不再详述。

第二，注重项目经验与积累，不看重学历与工作年限。做一个项目，犹如打一场战役，身经百战，积累下来的成功经验可以让工作更得心应手，失败经验可以避免走很多弯路。

第三，能够在1~2个以上技术领域精通。所谓术业有专攻，能够在某几项技术领域做到精通的人，相信对于新的技术领域或者从未有过相关经验的新项目，也能够轻松胜任，做到尽善尽美。

第四，关注应聘者的知识广度。如今的项目，已经告别个人英雄时代，讲究团队的协作。知识面越广，尽管在非专攻领域的深度可能不够，但是，知己知彼，可以站在一个更高的角度上看问题，这对于团队协作开发、项目融合的益处是显而易见的。

第五，具备良好的领悟能力、思考能力、设计能力、创新能力。基础技能知识不够可以学习，经验不足可以积累，技术不精通可以钻研，知识面不广可以开拓，但要培养这四项能力，是一件非常困难的事。要打造一支优秀的团队，这四项能力不可缺少，它们的重要度甚至超过以上的四方面要求。

我不会经常去问固定的问题，但所问的问题，几乎都跟以上的这些方面相关。

许式伟： 您曾尝试开放自己的程序代码吗？您对中国国内开源社区的现状有何看法？

张宴： 是否开源自己的程序代码，跟所在公司或部门的性质有着密切的关系。如果是在研发驱动型企业或部门，程序代码是公司生存的命脉，需要与竞争对手拼技术和保持技术领先的优势，因此，很难支持开源事业。反之，如果是在运营驱动型企业或部门，技术是用来提高运营质量、运营水平的工具之一。将纯粹的技术代码或产品，从公司的业务产品中提取出来，进行开源，可以按照开源产品的要求，提高公司内部技术产品的规范化、标准化，还可以引用更广大用户的使用、反馈和意见，解决未发现的潜在Bug，改进代码质量，提升技术水平。对于提高运营质量、运营水平来说，益处多多。我也尝试开源自己的一些代码，例如简单消息队列服务HTTPSQS（http://code.google.com/p/httpsqs/）、MySQL HTTP/REST客户端MySQL-UDF-HTTP（http://code.google.com/p/mysql-udf-http/），同时，也鼓励团队成员尝试开源，例如刚才提到的自动同步软件Sersync。

国内的开源社区在不断壮大，很多知名互联网公司都开源了自己的一些产品，但大多数还只停留在开源产品的使用、技术交流、汉化层面，真正参与到开源产品编码中的人还是较少，很多开源产品最终还是由原作者或原公司团队维护。国内开源社区的道路仍然漫长。

一分钟先生
Mr. One Minute

我的2010年

走过2010年，您都有哪些收获和感悟？本期一分钟先生栏目邀请IT业内知名人士，分享他们的2010年。



吕建伟

《走出软件作坊》图书作者

2010年我离开了工作生活将近11年的北京，全家迁移到了深圳，面临的新工作平台却不是《走出软件作坊》中描述的三五个人、十来条枪，而是200多人的研发团队；而且明年就要突破300人，在未来的3年内更是会拥有自己的研发大厦。这么大一个团队应该往哪里去？要产生怎样的业界影响力？这是让我深思的问题。因此我买了大量关于如何打造大学的书籍，通过思索“为什么那么多革命性的创新会出现在斯坦福或哈佛”等问题获取灵感。

我深深知道如果一个研发团队不具有校园创新与研究氛围，这么一大群人只会成为乌合之众，甚至更会成为效率极低的官僚集团。所以我做了以下几个关键性事情：第一，**重申一流研发中心的目标**，包括梦想、使命、愿景、精神、理念和未来三年技术架构发展规划；第二，**专业线清晰分工**，前端架构部门、应用架构部门、平台架构部门、数据架构部门，专家、持续专注，特种部队突破团队和大规模研发团队分离、轻快预研模式与正规版本生产模式分离；第三，**技术大讲堂**，每两周专门抽出半天专门用于技术分享技术交流；第四，**拿真实数据说话**，鼓励和推动大家从日常细节中找点滴问题，打开细节深入剖析研究，靠真实的代码、数据库数据统计说话，不要靠经验侃侃而谈；第五，**工具化**，把规范做进软件中、做进工具中，用软件提升了我们客户的管理能力，我们也得给自己制造工具，来提升我们自身的生产能力与生产质量；第六，**再敏捷一些**，出了异常，快速决议机制，以后再重现异常，就按机制办事，少开会，多每天随时碰面，多谈异常少谈正常；第七，**沟通**，多和基层沟通，多和一线销售、实施、疑难杂症支持沟通，多和市场营销沟通，多和老板沟通，多和中层沟通，形成强有力的组织脊梁，并反复问自己：我能帮你做些什么？



刘大卫

街旁网CEO

记得美国风险投资公司Y Combinator合伙人保罗·格拉汉姆说过，北京有着出色的技术人才、资金、适合创业团队成长的一切可能性条件：顶尖的院校、许许多多的创业型团队和VC……北京，在创业者的眼中俨然就是下一个硅谷。2010年，在这里，我和我的团队开始了在LBS这个新兴领域的市场角逐。我深知LBS是一种更本地化的用户体验，所以面对这样一个广阔且充满挑战的市场，一位了解SNS、懂技术、明白什么是LBS并看好项目前景的合作伙伴是非常重要的。幸运的是，2009年年底，我结识了我的合伙人、街旁网联合创始人杨远骋。2010年2月中旬，我们正式开始合作，街旁的想法也有了雏形。

2010年可以说是中国LBS元年。今年年初，我们看到了行业中涌现了基于LBS的很多有趣的移动社交服务。面对快速发展的中国移动市场领域内如此多种类的手机和平台、基数庞大的用户群体，我们深知必须加快行动的步伐。与此同时，我们由衷地欣赏Foursquare、Gowalla、Brightkite等一些美国的创业公司，并从它们身上学到了很多宝贵的经验。从美国到中国，从eBay到淘宝，从Foursquare到街旁，我们看到世界各地的人们其实是有着一些相似的需求，而我们做到了去表达和用独特的方式履行了这些需求。在街旁网正式上线运营的七个月的时间里，我们受益于开放式平台以及合作伙伴给予的积极支持。2010年12月28日，我们将进行街旁网新功能的发布。我们不是Foursquare在中国的简单复制品，而是真正基于中国用户需求、不断提供更好服务和用户体验的街旁网。2010年只是我们的一小步，对于即将到来的2011年，我们做好了迎接更多挑战的准备，也充满了对未来的美好希冀。

关注IT人自我管理、职业发展，广邀经验人士排疑解惑。

花一分钟阅读，还你一个明白



叶忻

架势无线CEO

2010年，给我印象最深的事情有那么几件。

第一，无线互联网在沉默了大概3~4年之后，突然发力，变成目前中国一个新的经济增长点和推动力。基本上，大家每时每刻都在讨论这个话题，从手机厂商和移动运营商，各家都在举办各种各样的无线互联网应用开发大会，而且也都舍得花大本钱办这个事情。我认为这是一个非常好的势头，可以说，我们只有想不到的，没有做不到的。况且，现在手机上的很多应用其实还是需要的。

第二，可能与我个人的行业不太相关，那就是我认为今年中国微博的广泛应用，必将会对整个中国的生活和传播都有很大的积极作用。如果五年以后回过头来再看，这样的感触应该会更深。

第三，毫无疑问，就是近来闹得沸沸扬扬的360与QQ的争论。通过这个事件，提高了中国所有互联网用户和电脑用户对隐私保护和反垄断、公平竞争这些问题的探讨。实际上，只通过互联网这个行业，就已经让我们去思索了很多问题：怎样才能保证有创新？我觉得互联网行业就是一个挺好的可以突破的创新方向，然后通过这个引申到其他行业以后，你就会发现很多问题。其实在创新方面，发达国家是非常成熟的，他们不只是为了简单的小进步，而是为了能够实现真正的创新。最终目的还是为了消费大众服务。



黄晓东

移动立方创始人

大家普遍认为2010年是移动互联网之年，而在我看来，移动互联网也就是互联网，只不过终端扩展搬到移动设备。这其中，有一个明显的现象，那就是Android手机数量开始增速。在美国，Android手机的出货量比去年增加了差不多十倍，虽然中国这方面没有确切的数字，但增长是定局，只不过可能没有增长这么多倍。而明年随着运营商的推动，相信会有迅猛的增长。

究其原因，就在于运营商，只要没搭上苹果这条船的，肯定都会力推Android，所以在数量上超过苹果是必然。但是这并不意味着，Android一定会战胜苹果。这其中有几个因素。第一，苹果设备的价格目前偏高，就算在美国也是同等设备里面偏高的。在中国，更有点超过普通老百姓的消费能力。实际上，苹果的设备有非常大的降价空间。它号称一台iPhone4可以赚到300多美元，成本也就是一百多美元，所以说如果它要愿意的话，价格一点都不会输给Android，只是看它想不想而已。价格战可能不符合苹果的性格基因，或者换句话说，现在其实Android对苹果还并不构成绝对威胁。第二，Android现在身逢乱世，尽管设备数量是增长了，但与平台价值相关的还有多种因素。由于历史遗留因素的关系，Android设备的API、软件分发渠道、计费等方面，都对开发者有严重的阻力，因为开发者自己没有能力去解决这些问题。正基于此，Google和众多厂商已经在合力解决这些问题。

正是看到这些大势，我从微软出来创业，可以说正好是个时机点，不早也不晚。



潘爱民

任职于盛大创新院，兼任盛大网络COO专家顾问

2010年我的一个收获是，完成了《Windows内核原理与实现》一书，作为对过去几年研究Windows操作系统的一个总结。在这一年我也顺利地完成了一次职业转换，加盟盛大创新院，开始新的职业生涯。

我一直觉得软件是可以做得很精致的，精致得像艺术品。好的软件题目，小到一个基础算法和函数，大到一个系统平台，值得做一遍再做一遍，没有最好，只有更好。从事软件开发的人（如我），多一份追求，便多一份精进的机会。

模型即软件，建模即开发

——顶点软件LiveBOS灵动业务架构平台

■ 文 / 许欣友

回忆：软件开发模式演进与发展

自上世纪40年代出现软件以来，尤其是1955年第一家软件公司CUC在美国成立，软件产业开始独立于硬件产业，并走向高速发展的道路。如今软件在生活中亦无处不在，从核工厂到航天飞机、卫星通信，再细微到股票交易和移动电话，均随处可见软件应用的身影。软件已经和我们的生活和工作密切相关。

软件产业的一个基本现象，就是基础平台与工具技术的更迭和变革愈演愈烈，完全超出人们的预期和意愿，以及个人所能应付的范围。在短短十多年间，基础技术经历了从面向过程到基于对象，从基于对象到面向组件，从面向组件到面向对象，从面向对象到面向服务等的重大变迁。就软件开发模式而言，主要经历了过程语言编程模式、面向对象的编程模式、面向组件的编程模式和平台开发模式四个发展阶段。从操作系统平台的出现开始，我们已经可以通过调用操作系统的API编写一个个漂亮的软件了。随后J2EE和.NET的出现，让我们构建复杂的分布式应用系统容易了许多。当然我们构建复杂的业务软件仍需要了解大量的技术细节和漫长的开发周期。现在，随着互联网的出现及技术的发展，“网络化、服务化、平台化”已经成为软件业发展的最新趋势。尤其是为了进一步降低在软件基础架构平台开发复杂应用系统的难度，一个更接近应用软件的新层级—— 业务架构平台，开始快速发展，而且自2002年以来，已经引起了业内的广泛关注。

那么业务架构平台究竟是一个什么样的东西呢？它和J2EE、.NET 等基础技术平台有什么不同？它在构建复杂的业务软件过程中能带来什么好处呢？接下来让我们一起来认识一下基于业务的软件基础架构平台。

创新：模型即软件，建模即开发

业务架构平台是从业务视角来看软件的研发和构建。它是以业务导向为驱动的、可快速构建应用软件的架构体系，是一个新出现的软件层级。业务基础平台包含了平台集成运行环境和与之配套的快速开发工具，更完善者还配有管理控制台系统。与操作系统平台、软件基础架构平台相比，业务架构平台和用户的管理及业务相关度比较大，是应用软件开发的通用基础平台。



图1 计算机各平台关系图

总的来说，业务架构平台类似一个优秀的建筑架构设计师，一切都围绕用户的应用需求，“业务导向，技术无关”，并从整体、全局和发展的角度去设计、搭建适合用户的信息系统平台，使得管理与业务人员参与应用软件的开发成为可能。

一般的软件开发，不仅需要在完成业务需求分析与系统设计后，进行数据库的设计（数据库设计人员参与）、业务逻辑设计（J2EE、.NET程序员），以及Web页面设计（JSP、ASP技术人员、界面设计人员），而且还要经过测试、部署，才能最终完成软件的开发工作。一旦遇到需求的调整，仍需循环以上过程。所以整个周期、流程都比较长。

目前还有一种软件开发策略是“组件化”开发模式，它通过预先开发大量的组件，来简化一些重复性的编程过程。这样，虽然可以减少部分重复性的软件开发工作，但如果组件封装的粒度太大，适应性就弱化；如果太小，则无法体现



图2 LiveBOS的产品构成与关系图示

应用组件模式开发的效率。而且组件往往只能在界面层面或单一的应用服务器层面发挥作用，在整个大型应用软件中，涉及到数据库、应用服务器、界面等环节，组件很难在深层次上发挥价值。

而由福建顶点软件股份有限公司研发的LiveBOS灵动业务架构平台（以下简称“LiveBOS平台”），它的目标正是为了降低技术复杂度对开发和应用的影响，支持构建随需应变的应用，让开发人员从琐碎烦杂的代码中解放出来，着眼于更高层次的业务模型和业务逻辑，开发复杂的业务软件。

LiveBOS平台重大的创新在于用对象化的业务建模与业务流程建模，来替代许多同类软件那种组件式的软件开发模式。 它不生产任何的Java或C#之类的编程语言，而是依托LiveBOS的业务中间件服务器（LiveBOS Server），直接解析业务模型，实现软件功能。

LiveBOS系统由三个相对独立的产品构成：运维支持支撑平台LiveBOS Server、开发集成环境LiveBOS Studio以及运维管理工具LiveBOS Manager。

LiveBOS平台采用了面向对象的业务分析与建模技术。在LiveBOS平台中，通过设计并输入对象模型，就可以建立完全贴近业务需求的实际应用，期间完全屏蔽了技术实现的

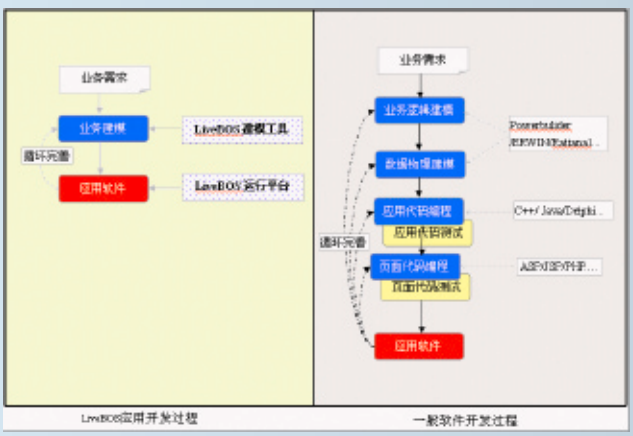


图3 LiveBOS和一般软件开发过程比较

细节。伴随对象模型建立的完成，即刻就能实现软件的功能。如果需求调整，只需直接调整相应的业务模型，就能即刻实现软件的变更。这样的开发模式，极大提升了软件的研发效率。据统计，在实际的应用开发中，它使得软件产品的研发周期平均减少了70%。这样，采用LiveBOS平台开发，不仅仅省掉大量的编码工作量，还大大缩短了系统建设周期，提高了稳定性。

因此基于LiveBOS，开发者只需要基于业务和管理的层面，而非技术的层面来理解、设计、架构和集成企业的信息系统（基于业务层面是指开发人员只需要描述企业的组织机构、业务流程、业务信息、业务资源、业务逻辑、业务事件等业务内容，而不考虑技术层面的东西），就可以实现各类基于Web的高层次的信息化应用。而且，用户可以随时在运行中重新定义或调整业务模型，从而达到使自己的信息系统完全贴近不断变化的业务需求……这也是“灵动”的价值体现。

价值：变人月神话为人天现实

LiveBOS 为复杂项目开发所提供的丰富价值，具体表现如下。

- 业务驱动，敏捷开发，持续改进

“模型即软件，建模即开发”是LiveBOS的最重要的特征。它可以从根本上解决开发过程中的“需求困境”。LiveBOS支持敏捷开发模式，它允许以在开发过程的任何阶段提出需求改进，通过快速循环改进模式，提升软件功能。

- 变人月神话为人天现实

整个LiveBOS应用的开发过程，就是业务建模的过程，它完全替代了一般开发模式中系统设计、数据库建模、程序设计、代码测试等工作量十分大，并且需要大量循环完善的工作，与一般管理软件的开发周期相比，可以减少70%以上的时间。我们可以通过下述案例来进一步体会LiveBOS快速建模在实际应用中带来的价值。

LiveBOS对于大型、复杂项目的价值，在上海工艺美术

职业学院中得到了充分的体现。

上海工艺美术职业学院是经上海市人民政府和国家教育部批准的独立设置的艺术设计类特色高等院校，是中国唯一一所艺术设计类示范性高职院校。为了体现其艺术院校的管理特点，并提高校园管理的高效性、增强整个学校不同业务环节的协同性，上海工艺美术职业学院选择了顶点的LiveBOS平台，作为新一代的信息化系统基础平台，彻底改变了原来各个应用软件零散、低效、信息与流程无法协同的问题。

作为新一代信息化建设的第一期项目，上海工艺美术职业学院的“数字化教学资源管理平台（一期）”项目从2009年6月15日开始，在接下来的三个月的时间里，经过需求分析、业务建模及测试，到2009年9月时，招生管理信息子系统、迎新管理子系统、学生收费管理信息系统均已开始上线，而其他子系统（包括教务信息管理系统、学生工作管理系统、人事信息管理系统、教材管理信息系统、教学资源管理系统、信息门户管理系统、办公管理系统、教学质量监控评价系统等）也于2009年12月基本开发完成并上线运行。同时，我们在充分调研的基础上，并结合我们数字校园管理系统的实施情况，总结出基于LiveBOS平台的软件开发模式和传统开发模式的主要区别：

上海工艺美术职业学院数字校园管理系统（一期）			
项目		基于LiveBOS平台的开发模式	假设采用传统开发模式
开发环节		需求分析（约50%） 业务建模及测试（约50%） 生成应用软件	需求分析（约30%） 业务逻辑建模（约10%） 数据物理建模（约10%） 应用代码编写及测试和页面代码编写及测试（约50%） 生成应用软件
人员配置	项目经理	1	1
	需求分析师	5	6
	业务建模工程师	7	0
	数据库工程师	1	1
	程序员	0	12
	测试工程师	3	5
	项目实施工程师	5	5
	文档及美工	1	1
开发周期		大约6个月	大约10个月
开发成本		约60个人月	约110个人月

注：以上是在全新项目开发的情况下得出的开发对比表。如果是二次开发，由于LiveBOS平台开发的软件具有高度复制性的特点，基于LiveBOS平台的开发模式比传统开发模式的效率又会有较大幅度的提升。

● 应您所需，随时而变

基于LiveBOS平台，所有的业务管理模块，都可以根据企业的不同组织模式、业务特点、流程需求，进行快速调整，这样企业就可以得到完全个性化的、体现企业特色的应用系统，而不是一个僵化的、需要“削足适履”的应用。同时系统可以随企业业务的发展变化进行动态调整，实现系统与业务的同步发展。

LiveBOS 灵动业务架构平台已经能够为企业的信息系统提供一个完整的解决方案，并使业务开发人员无需关心底层的技术实现细节，无需了解J2EE的相关知识就能快速构建出一个基于J2EE体系的应用系统来，在此期间无需编写一行的Java或者JSP代码，几乎是零编程的方式，便可使业务开发人员完全从技术细节解脱出来，全心地关注业务描述。

LiveBOS作为顶点软件致力于服务企业信息系统建设的业务基础平台，除了解决当前应用系统本身的业务功能之外，也为信息系统的扩展提供了广大的扩展空间。我们可以看出，利用LiveBOS构建业务应用，系统复杂度有了数量级的下降，系统设计和开发可以合二为一。这结合了传统IT解决方案的优点，不仅可以根据企业的特点量身定做，而且拥有随着企业业务发展而不断进化的能力。

总之，LiveBOS致力于提供一个用于管理变革的长期平台，为软件持续工程努力提供帮助，并通过完整的平台使开发者在项目最初和以后都能进行业务流程分析、设计和配置。采用基于LiveBOS平台进行企业应用开发，将提高企业IT系统对业务变更响应的敏捷度，真正实现随需应变的应用，降低IT系统开发运行的风险和成本，提高企业客户满意度。

作者简介 | 许欣友



上海工艺美术职业学校信息处处长兼校企合作处处长。

Java的命运

■ 文 / Peter Seibel　译 / 郝培强

本文是Common Lisp专家Peter Seibel对Google公司首席Java架构师Joshua Bloch的访谈，谈到他所遇到的最糟糕的Bug以及Java的命运。

最糟糕的Bug

Seibel：我们聊聊调试吧。你遇到的最糟糕的Bug是什么？

Bloch：提起Bug我立马就想到了一个，这个Bug很严重，而且很搞笑。那是90年代初，我在匹兹堡的Transarc公司工作时。我在很紧的工期下提交了一个事务共享内存的实现。我在限期内完成了设计和实现，甚至还在过程中做出了几个可重用的组件。但是这么匆忙地写了很多新代码，我还是挺担心的。

为了测试这些代码，我写了一个叫做“乱撞”的很长的程序出来。它运行了大量的事务，每个事务又包含了嵌套的事务，嵌套到可以嵌套的最大深度。每个嵌套事务都可能会加锁，以递增的顺序读取共享数组里面的几个元素，对每个元素都加入点东西，保持数组中所有元素的和为0，还是不变量。这些事务要么提交，要么取消，如90%的提交，10%的取消，其他比例也可以。多个线程同步运行于这些事务之上，长时间地访问数组。因为我测试的是一个共享内存机制，所以我同时运行多个有多个线程的“乱撞”程序，每个有自己的进程。

在一般的并发级别下，“乱撞”轻松过关。但是当我真正调高并发级别时，我发现“乱撞”偶尔，仅仅是偶尔，无法通过一致性检查。我不知道这是怎么搞的。这只能是我的错，因为新代码都是我一个人写的。

我花了大约一个星期，痛苦地为每个组件写了彻底的单元测试，所有的单元测试都通过了。然后我为每个内部数据结构写了详细的一致性检查，这样我就可以在每次变化后调用这些一致性检查，直到测试失败为止。最后，我终于发现一个底层的一致性检查失败了，这个问题无法重现，但是某种程度上可以帮助我分析问题出在哪里。最后，我得出了确实的结论——我的锁根本不工作。两个事务锁定、读写同一个值的时候，产生了并发的读—修改—写回操作，而后一次写入毁掉了第一次的写入。

我编写了自己的锁管理器，所以我怀疑是它出了问题。但是锁管理器轻松地通过了测试。最后，我觉得问题不在锁管理器，而是它依赖的互斥体的实现！那时候操作系统还不支持多线程，我们需要写自己的多线程包。原来负责互斥体代码的工程师，不小心把我们的Solaris的线程实现中的lock和try-lock的汇编代码的标签弄混了。所以，每次你以为你在调用lock的时候，其实调用的是try-lock，反之亦然。也就是说当真的有争用发生的时候——在当年其实是很罕见的——第二个线程直接就进入了第一个线程的临界区，因为第一个线程也没有锁住。搞笑的是，这也就是说，整个公司几个星期都在运行没有互斥体的程序，而且谁都不知道。

Knuth有句关于测试的名言，Bentley和Mcllroy的精彩论文“Engineering a Sort Function”中曾经引用过，大概意思是说，做测试时，要不惮以最大的恶意来推测所要测试的代码的错误。做这些测试的时候，我就是这么做的。但是这样会把所有东西纠结在一起，更难找到Bug。首先，并发的时候很难这么做，往往完全无法复现场景。其次，到最后可能会发现你的核心假设是错的。喜欢喊“耶，这语言出错了”或者“系统出问题了”是新手干的事儿。但是在这里，我依靠的基石——互斥体，确实出问题了。

Seibel：也就是说Bug不在你的代码中，但是同时你只能对你的代码进行彻底的单元测试，因为你没有别的办法，只能去检查自己的

代码。你觉得这些测试是不是可以，或者说应该让互斥体代码的作者来写，这样你就不用浪费一个星期，节省了一半的测试量，而且也可以找到这个Bug。

Bloch：给互斥体代码加一个好的自动化单元测试肯定可以避免让我遭受那些痛苦，不过注意那可是90年代初。我想都没想过要抱怨那个工程师没写个好的单元测试。即使是今天，为并发工具写单元测试还是一种艺术形式。

Java的命运

Seibel：既然你说到这里，Java是否逃脱了灭亡的命运了？它变复杂的速度是不是比变好的速度更快呢？

Bloch：这个问题不好回答。具体说来，Java 5加入了比我们设想的更多的复杂度。将泛型特别是通配符加到语言中到底有多复杂我也说不好。我得为有功劳的人说句话，Graham Hamilton真是了不起，那时候他就想明白了一切，而我不明白。

当你改进一个成熟语言的时候，你必须更加仔细地考虑能力和复杂度之间的平衡。

有趣的是，他抗争多年，希望阻止泛型进入Java语言中。但是在泛型被成功地阻挡在Java外的这些年里变体的概念，也就是通配符的隐含意义流行了起来。如果它们来得更早，没有变体，也许我们现在可以有一个更简单的、更容易跟踪的语言。

引入通配符有实际的好处。子类化和泛型之间根本就是阻抗不匹配的，通配符尽力在弥合这种不匹配。但是这么做又显著地增加了复杂度。有些人认为在声明空间，而不是用户空间，变体是更好的解决方案，但我不太相信这一点。

这仍旧悬而未决，因为它们都还没经过在真实世界里海量的程序员们的测试呢。一些语言经常只在小范围内获得成功，人们会说：“噢，这些语言很不错，只是可惜没有成为世界范围成功的语言。”但是这往往是有原因的。希望使用Scala或者C# 4.0，这样的声明空间变体的语言可以彻底解决这一疑问。

Seibel：那么是什么推动Java引入泛型呢？

Bloch：没看起来那么精彩啦，我们的新闻报道是可信的。我的思维模式是，“嗨，集合多半都应该是同质的——一组字符串，一个从字符串到数字的映射，等等。而现在默认情况下集合是异质的：它们都是对象的集合，取出时都需要类型转换，这简直是胡闹。”如果我可以告诉系统，这是一个从字符串到数字的映射，它会帮我做类型转换，而且会在编译期间帮我盯着，防止我做错什么，那不是挺好的吗？它可以抓到更多的错误——它可以包含高层的类型信息，看起来是件好事儿。

我认为泛型和其他加入到Java 5的语言特性一样，我们只是让语言去做以前我们要手工去做的事情而已。某些情况下我坚信：foreach就是好。它所做的就是对你隐藏遍历器和索引变量带来的复杂性。代码更短，概念也不复杂。从某种意义上说，它的概念更简单，因为我们为数组和其他的集合创建了这种伪多态机制，你可以遍历一个ArrayList或者一个数组，而无需关心你遍历的是什么类型。

这种思想不能适用于泛型的主要原因是，它是对已经很复杂的类型系统的大扩展。类型系统是很微妙的，修改它们可能对语言带来深远的、难以预期的影响。

我认为得到的教训是，当你改进一个成熟语言的时候，你必须更加仔细地考虑能力和复杂度之间的平衡。而且，实际上，复杂度跟语言的功能数量间至少是平方级关系。为一门老语言加上了一个新的功能，通常就意味着为它加入了一大堆复杂度。当一种语言已经达到或接近程序员理解能力的极限时，那么你加入任何复杂性进来都会加剧理解的难度。

语言更复杂后就会消失吗？不会。我认为C++早已超越了它的复杂度极限，但还是有很多人用它编程。可这实际上是逼人们只使用其中一个子集。所以我认识的每个用C++的公司都说：“对，我们用C++，但是用的是多继承，不用操作符重载。”有很多功能你完全不用，因为使用它们会造成代码太复杂。即使不得不用那些功能，我认为也实在没什么好处。那样的话，程序员就读不懂别人的代码，也就

不存在“程序员的可移植性”了。

Seibel：如果去掉泛型，现在Java会变得更好用吗？

Bloch：我不知道。我还是喜欢泛型。泛型能帮我找到代码中的Bug。泛型可以让编译器强制做一些限制，之前这些限制我只能放在注释中。另一方面来说，当我看到那些疯狂的参数类型相关的错误信息，当我看到像class Enum<E extends Enum<E>>这样的泛型类型声明时，我就会想，显然泛型的设计还没成熟到可以放到Java中的水平。

我们总是太乐观，然后搬起石头砸自己的脚。所以我们说：“耶！我们当然可以把泛型放到Java中。在CLU的时候我们就知道泛型了。这技术25年前就有了。”最近我听到关于闭包的类似言论，不过那是50年前的技术了。“噢，闭包很简单，不会给语言加入任何新的复杂性。”

嗯，没错。但是我觉得我们从泛型这件事儿得到了教训。在你懂得这个改动会对概念层面带来什么影响之前，在你可以确保软件行业从业人员可以高效地使用新特性，而且这一新特性会让他们活得更好之前，你不应该给语言加入这一特性。

如果早知道程序员们对泛型是这个反应，我们肯定不会把它加到Java里。这是不是说我们就完全不会搞泛型？不，我不这么认为。我认为泛型确实很好。主要是因为大多数集合是同质的，而不是异质的，同质的集合处理起来是比较方便的。多数情况下类型转换都不合适。转换可能会失败，而且让你的程序不再优雅。我想你知道这是什么集合，它应该自动符合你的这些需求。但是，是不是这就意味着你应该承受我们现在承受的这种复杂度？不，我想我们只是没有处理好泛型。

Seibel：关于泛型有来自于用户的压力吗？有人抱怨泛型的缺点干扰他们写程序了吗？

Bloch：有没有工程师大骂泛型的缺点？不，没有，他们没有抱怨过。如果因为泛型简洁就把它们加进来，那我会内疚的。因为当时我们以为这么做是对的。

有人说，很多工程都是扯淡。有人要求我们加入foreach吗？没有。他们没有要求我加入。但是我就知道这是应该做的。我对了——每个人都喜欢它。但是我觉得我们行业内的一大问题就是，在工程领域，做一个东西，仅仅因为它简洁，仅仅因为它是一个好的工程项目，等等。如果你不能解决真实用户——在这里就是Java程序员——的真实问题，那么你就不应该加入新的特性。

James Gosling曾做过一个非常了不起的演讲——“Java的感觉”。他说，给Java加入任何东西之前，都需要三个真实的用户。不应该因为一个东西简洁就把它放进来。



Joshua Bloch，现任Google公司首席Java架构师。之前他在Sun公司工作，领导并实现了Java 2中的Java Collection Framework，还参与了Java 5发行版中几项语言附加特性的设计。Bloch在卡内基梅隆大学获得博士学位，期间他参与设计了Camelot分布交易处理系统，后来演变为Transarc公司的产品Encina，而他则是Transarc的资深系统设计师。他编写的Effective Java一书获得了2001年Jolt大奖，他还与人合著了《Java解惑》和《Java并发编程实践》。

但是人们就是想把什么东西都放进去。工程师是做什么的？他们就是写代码的。而当他们写一个库，或者一个语言的时候，他们就是想放各种东西进去。你需要他人的参与，需要指导的声音，需要这些东西来帮助你完成产品，帮你在放与不放之间做出最好的权衡。因为你可以放进去的东西总比你应该放进去的东西多。那么是不是说所有的这些东西都不好呢？那也不是。只是你需要做出决定，某些东西是不应该放进去的。

思考Java带来的编程经验

Seibel：思考Java的设计并实现它，是否让你学到了什么跟编程有关系的东西？

Bloch：我学到的东西太多了。比如我知道了即使是想把一个很小的程序写对也是非常难的。我把这个想法发表在了博客里，题目是“几乎所有的二分搜索和归并排序都是错的”。认为自己程序是对的就是在愚弄自己。程序里有大量没解决的Bug，当然是不对的。多数情况下，程序里的Bug都不少，它们只能免费完成任务。

我知道，既然写正确的程序那么难，我们就应该尽力去帮助大家。所以能减少Bug的所有东西都是好的。这就是我是静态类型和静态分析的信徒的原因，任何可以减少某个特定类别Bug的东西都是非常好的，任何可以让程序员的工作更轻松的东西都是好的。

我更加确信有好的API文档是很重要的。人们很少提及Javadoc对这个平台的成功所起的作用。好的API文档永远都是Java文化的一部分，也许是因为Javadoc从一开始就存在吧（译者注：所以人们低估了Javadoc的作用）。

我一直信奉“简单就是美”这句话，现在更是如此。我不断看到更复杂的东西最终被证实是有害的，只是有的时间长点儿，有的时间短点儿。我设计的时候，会仔细看着我的“复杂度计”，一旦复杂度要到红线了，就需要重新设计了。

偶尔我会遇到不相信这些的人们，他们会说：“Josh你太傻了，你怎么就是不明白；这才是应该做的，可惜你就是搞不懂。”我就是不信这些。我觉得事情一旦复杂起来，那么一定有什么地方错了，也许到了寻找更简单的方法的时候了。

Tony Hoare的图灵奖获奖感言中有一句充满了大智慧的话，讲的是设计一个系统的两种方式：“一种是尽量简单，这样显然不会有什么问题；另外一种是，尽量复杂，这样没什么问题会很显然。”

后面的内容同样饱含智慧，但是知道的人不多：“第一种方法其实更难。它需要从复杂的自然现象发现简单物理规律的那种技能、投入、洞察力，甚至是那种灵感，同时还需要你能接受你的目标受限于物理、逻辑和科技的约束，以及在目标间有冲突的时候可以妥协。委员会无法做到这些，除非已经完全来不及了。”

Seibel：你是否想过在职业生涯中再次更换你的主要语言，还是准备退休前一直做Java?

Bloch：我自己也不知道。我从C语言转向Java有点突然。从研究生毕业时起，到1996年，我主要使用C语言编程，然后一直使用Java直到现在。我已经预见到我可能要更换到其他语言了。但是我不知道是什么语言。也许它还不存在。我觉得产生一个新编程语言的时机已经成熟，但是同时我又觉得平台的惯性也比以前更大了。现代的平台不仅仅是一个语言和一些库，它包括很多工具，是一个虚拟机，一个庞然大物。创建一个完整的新平台的前景比以前更不乐观。

我不知道将出现什么。但是我认为如果改变我的主要语言是对的，那么我就会这么做。我想尽力保持开放的心态。我想尝试更多的语言。我最近没时间做，但是以后还是会做的。

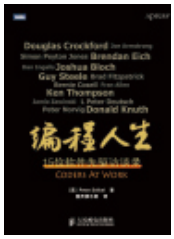
Seibel：列出几个你想尝试的语言吧?

Bloch：我想试试Scala，虽然我怀疑它是否能成为未来的新宠。我很崇敬Martin Odersky。我觉得他写的语言中有很多精妙的想法。但是我同时也认为他加入了太多复杂的东西，太学术化了，所以很难取得世界范围内的大成功。当然我还没权利去评价，因为我还没学习过。

我还想用用Python。Scheme不是新生物，不过我也想试试。我想花几个月，跟我儿子一起过一遍《Structure and Interpretation of Computer Programs》一定很有意思。每个人都说这是一本伟大的书。我已经买了，算是开了个头。看完它需要一些时间。我想这就是我目前想学的。

Seibel：现在很多人在讨论我们写程序的时候，如何能把未来的多核CPU的优势利用起来。Java显然是第一个内建多线程机制的主流语言。你觉得Java的逻辑在多核的世界是否仍然可用?

Bloch：我想说得更深入一些。我认为Java是现有语言中最好的。但有趣的是，现在很流行谈Java是否即将死去。我觉得这基本上是扯淡。我认为现在最好的多线程构件就在Java里。我认为Java将迎来复兴。我不是说它是未来20年内最先进的，也不是说它是处理多核的最好方式。但是我认为从现有的东西来看，我们是足以傲视同侪的。



本文节选自人民邮电出版社北京图灵文化发展有限公司出版的《编程人生》一书。该书是当今15位大师级计算机程序员的访谈录，重点介绍了他们的编程感悟。特此感谢图灵公司授权。

【编者按】许多大师和业界专家都非常强调技术论文的学习，我们近期采访的图灵奖得主Chuck Thacker甚至将阅读历史性的经典文献作为自己的成功之道。2011年开始，我们特别开辟“论文研读”栏目，欢迎大家推荐有价值的论文。

分布式流式计算平台——S4

■ 文 / 马如悦

本文是作者在充分阅读和理解Yahoo！最新发布的技术论文《S4：Distributed Stream Computing Platform》的基础上，所做出的知识分享。

S4是Yahoo!在2010年10月开源的一套通用、分布式、可扩展、部分容错、具备可插拔功能的平台。这套平台主要是为了方便开发者开发处理流式数据（continuous unbounded streams of data）的应用。项目官方网站为：http://s4.io/。同时，S4的开发者也发表了一篇技术论文《S4: Distributed Stream Computing Platform》来介绍S4的设计。下面我们就来学习这篇论文。

开发动机

“We designed this engine to solve real-world problems in the context of search applications that use data mining and machine learning algorithms.” … “To process user feedback, we developed S4, a low latency, scalable stream processing engine.”

Yahoo！之所以开发S4系统，主要是为了解决它现实的问题：搜索广告的展现。搜索广告是当前各大搜索引擎的主要收入来源，用户发出查询请求，搜索引擎在返回正常结果的同时也会返回相关广告，而广告是按照点击付费。为了在最好的位置，放置最相关（也就是用户最有可能点击）的广告，各大搜索引擎使用了大量的数据挖掘和机器学习算法来进行相关性计算，以便提高收入，满足用户需求。其中很重要的一点就是要不断分析用户的点击反馈，以便捕获用户的行为。S4最初主要还只是用来处理用户的点击反馈。

“The streaming paradigm dictates a very different architecture than the one used in batch processing. Attempting to build a general-purpose platform for both batch and stream computing would result in a highly complex system that may end up not being optimal for either task.”

那么Yahoo!为什么没有选择Hadoop来处理呢? MapReduce系统主要解决的是对静态数据的批量处理，即当前的MapReduce系统实现启动计算时，一般数据已经到位了（比如保存到了分布式文件系统上）。

而流式计算系统在启动时，一般数据并没有完全到位，而是源源不断地流入，并且不像批处理系统重视的是总数据处理的吞吐，而是对数据处理的latency，即希望进入的数据越快处理越好。

当然，现在也有很多基于Hadoop系统来处理流式数据。一般有以下几种方式。

- Micro-batch in MapReduce：就是把流式的数据按照时间或者大小形成小的静态数据，然后定期启动MapReduce来计算。
- Continuous MapReduce：Hadoop Online（http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-136.html）通过实现作业内的数据传输Pipeline和作业间的数据传输Pipeline，可以实现online aggregation和

continuous queries。当前MapReduce模型中，只有Map中间结果完全产生后，Reduce才会过来拖数据，等所有Map数据都拖成功后，才能计算。Hadoop Online实现了Map到Reduce间的数据Pipeline，使得可以在Map产生部分数据后，就可以送到Reduce端，以便Reduce可以提前或者定期计算。

- Dynamic add input：百度的一种实现，用来解决计算时数据还没有到位的问题。作业可以在数据还没有完全到位的情况下启动，当新数据累积到一定量时，通过一个命令行接口，向运行中的作业动态增加新的输入。通过这种方式，大大减少了处理大数据作业时等待数据到位的时间，在依次执行多个作业时，也会有时间收益。

在论文中，对类似于第一种的方式，分析了它的缺点。如果将数据流切成较小的data segment，就会增加启动作业的overhead，同时使得维护segment之间的依赖关系变得更加复杂；但如果切得较大，那么处理的latency就会比较长。

随着大量实时应用的发展，比如实时搜索、实时交易系统、实时欺骗分析、实时监控、社交网络等，都需要一个高度可扩展的流式计算解决方案。不同于原来的流式计算系统，S4主要解决的是高数据率和大数据量的流式处理。

设计假设和目标

为了简化设计，S4给出了下面的假设。

- Lossy failover is acceptable， 即一旦一个节点失败，会failover到另一个standby节点，但是会丢失原节点的内存状态。这也是为什么说S4是一个部分容错的系统。
- 节点不能动态增加和减少。

设计目标包括以下几个方面。

- 简单的编程接口。
- 高可用+高可扩展。
- 尽力避免Disk IO，而要尽量使用Local Memory，以便减少处理latency。
- 使用去中心化和对称架构，所有的节点的责任相同，方便部署和维护。
- 功能可插拔，使得平台通用化的同时，做到可以定制化。
- 设计要科学、易用和灵活。

S4的设计大量借鉴了IBM的Stream Processing Core（SPC）中间件的设计。只是SPC采用的是Subscription Model，而S4结合了MapReduce和Actors Model。

Event Stream

一个Stream是Events的序列流。每个Event是一个（K, A）数据，通过Event Type来标示其类型。K、A分别表示这种类型的Event的keys和attributes。key和attribute都是tuple-valued，即key=value这种元组值。下面给出一个event的例子：

```
EV: ClickLog                    → event type
KEY: product="search", type="online"
                                → keys
VAL: userid="123", ip="10.0.0.0",
cookieid="3"                    → attributes
```

Processing Elements

Processing Element（PE）是S4中的基本运算单元。一个PE通过下面四个组件来表示。

- functionality：实现PE的Java类和相关配置来定义。
- types of events：处理的event type。
- key：关心哪种key。
- Key的值：关心的key值是多少。

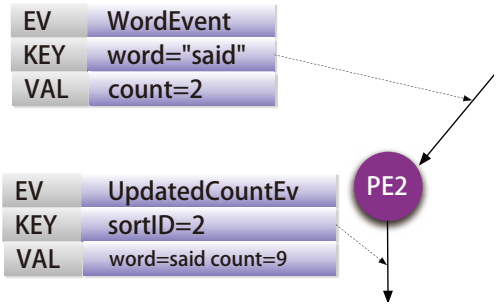


图1 PE处理举例

每个PE只负责处理自己所关心的event type，并且只处理自己所对应的key值的event。PE处理后可能输出一个或多个event。当平台处理一个key值时，会先检查相应的PE是否已经存在，如果不存在，会先初始化相应的PE，然后交由

这个PE进行处理。举例如图1所示。

在图1中，PE2负责处理相应的单词事件（WordEvent），主要逻辑是统计所关心单词的个数，然后输出给下游的PE。PE2所关心的event type为WorkEvent，所关心的key为word，所关心的key值为“said”。假如又来了一个WordEvent, key为word=“listen”，那么这个事件就不是PE2所关心的，所以平台可能会为“listen”值启动一个新的PE来处理。

有一类特殊的PE，即keyless PE（没有key和key值），这些PE会接收相应event type的所有event进行处理。这类PE主要用来作为S4 cluster的输入层（Input Layer），即外围应用会产生相应的event（keyless event），将这些event发到任何一个节点。而S4 cluster中的每个节点都会启动一个keyless PE，这些PE做简单的输入处理后，转化为keyed event，交给集群中的其他PE类型进行处理。

PE的逻辑主要由应用程序员来开发。

Processing Node

Processing Node是一个逻辑节点，负责监听消息的到来，对消息进行处理，然后通过Communication Layer将event在集群中分发。S4主要依据上面提到的event type和key/key值，对key值求hash，在集群中进行分发。关注的key集合通过配置文件来得到。对于需要处理的event，会交给PN中的Processing Element Container（PEC），然后PEC调用相应的PE进行处理。PN功能框如图2所示。

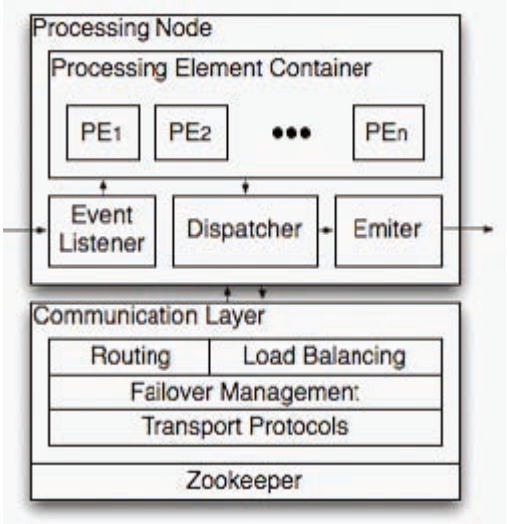


图2　PN功能框图

通过图2的设计，可以保证，对应于相同event type，key和key值的event一定会被路由到对应的PN。

底下的Communication Layer和Zookeeper共同完成了集群管理和自动failover功能。

编程模型

应用的主要任务就是实现一些相应的PE。PE一般提供如下接口供应用实现。

- processEvent()：用来处理每一个event，然后修改相应的内部状态。
- output()：框架会按照应用的配置定期的调用，以便向下游输出其他event。应用可以使用两种输出配置，一个是隔多长时间输出一次；另一个是隔多少event个数输出一次。

其他

论文中给出了一个Word Count的例子，大家可以仔细研究一下。在性能测试部分，论文总结了将S4应用到实际的CTR（Click-Through Rate）预估中的效果。在应用举例中，给出了S4在在线参数优化的应用。

随着大量实时计算需求的增加，分布式流式计算将会成为分布式计算的下一个主要研究重点，将会成为类似Hadoop这类MapReduce框架的有力补充。这一方向的工作还处在初级发展阶段，大家需要多加关注。

作者简介 | 马如悦



百度基础架构部高级工程师，自2007年加入百度，一直从事分布式存储系统和分布式计算系统的设计和开发工作。对Hadoop有较深入的研究，一直积极活动在Hadoop开源社区。

■ 责任编辑：董世晓（dongsx@csdn.net）

关于MySQL多机房部署那点事儿

■ 文 / 杨海朝

MySQL多机房部署中存在哪些问题？应该怎样合理解决？本文作者给出了精彩解答。

随着网站规模的扩大，用户对访问速度的要求越来越高，就近访问的原则越来越重要。对数据库的访问也一样，需要各地都有数据库，部署在各地的Web前端直接访问本地的数据库。站点能够大规模部署，要求程序具有可伸缩性，例如：要求系统采用异步策略和异步数据流，避免分布式事务以及抽象和虚拟化等问题，更涉及网站整体架构的伸缩性问题。系统架构方面我暂不做讨论，这里只讨论一下数据库层次的部署问题。

做MySQL多机房部署时，需要考虑几个方面：MySQL异步的数据同步再加上多机房之间的跨网传输引起的数据同步延时问题会给应用带来的影响；多机房部署中影响MySQL同步的因素有哪些，怎样缓解或解决这些问题；当某一个机房的所有Slave出现故障（例如程序Bug导致数据被清空）时，怎样更快地重建，使各个机房的业务能够在很短的时间内恢复；如何更好地监控多机房部署的MySQL服务；MySQL多机房部署中涉及的高可用性问题；在多机房部署能够多点写入的系统。下面就每个问题分别说明。

MySQL同步延时对业务的影响问题

MySQL的复制说到底是异步的方式，存在同步延时的问题。目前大多数应用都是集中写，本机房同步存在延时问题，跨机房的延迟问题更严重。对于实时性要求很高的业务，可以采用写完主库，第一次就从主库上读，避免同步的延时问题，或者通过一定的业务策略，写完之后给出一个几秒的等待跳转时间，来缓解同步延时对用户的影响，不过这种方式也只是从一定程度上缓解。对于读实时性要求很高的应用，不太适合多机房部署，如果考虑多机房部署，可以考虑采用专线来减少同步的延时问题，或一份多写，或多点写入。解决延时最好的办法还是本地写本地读。

多机房部署中影响MySQL同步速度的几个因素

MySQL同步依赖的因素比较多，如同步之间的网络环境、机器的硬件配置、SQL语句是否高效以及同步数据量的多少等。两边开启同步数据压缩，能从某种程度上减少同步的数据量，对减少网络带宽有一定的好处。

多机房部署的MySQL不适合更新数据量比较大的服务，如果更新数据量比较大，同步数据线路最好使用专线方式，保证同步足够的可靠和有充足的带宽资源。如果多机房之间采用树形结构的复制方式，最好控制在三层以内，中间不提供服务的MySQL服务采用blackhole引擎，减少同步的延时。关于SQL语句，建议不要建太多的索引，避免对插入速度的影响，同时注意更新和删除操作的索引问题，加快更新和删除速度。减少多机房之间传输的数据量，在两个不同的机房之间保证两条线路即可。当

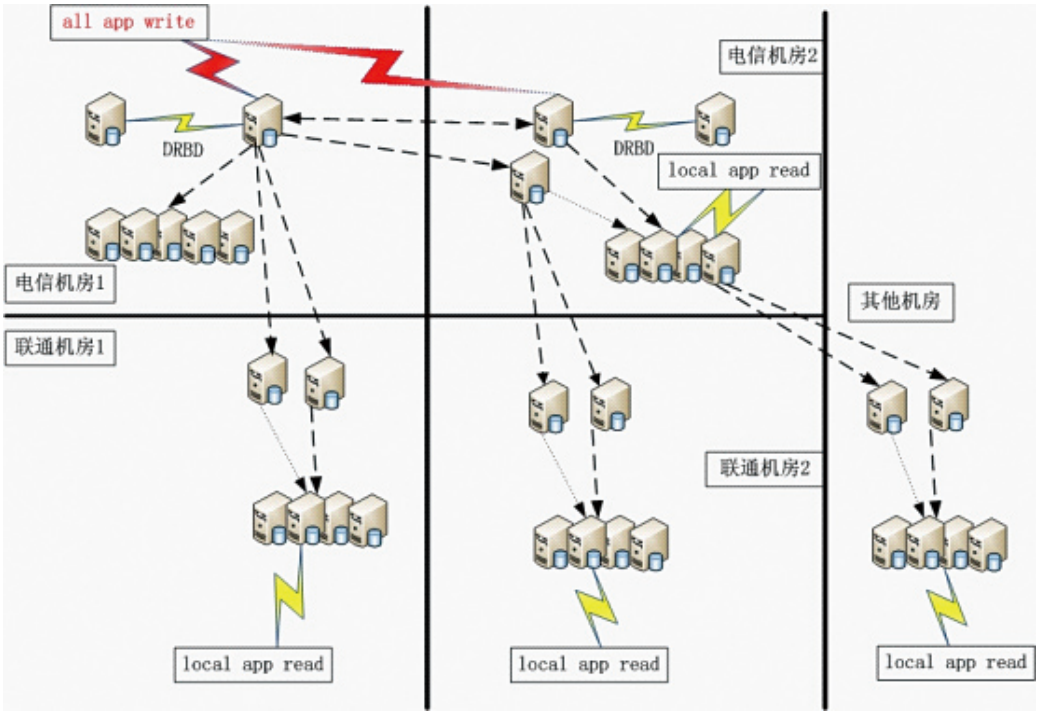


图1　MySQL多机房部署实例

每个机房都有多台Slave时，可以通过中继的方式，把一个机房的主库同步到另一个机房的中继，中继再向本机房的多个Slave机器分发数据的同步，减少机房之间传输的数据量。如果一个机房直接向另一个机房的Slave集群同步数据，那么有多少台Slave就需要多少条传输线路，但如果采用这种方式，就只需要一条线路，传输一份数据。这种级联的方式，从理论上说增加了延时，但减少网络带宽的占用带来的同步提升更为明显一些，特别是在多机房的大规模Slave集群部署上。中继服务器因为使用blackhole引擎，因少了一步写入数据库文件的操作，对提高同步速度非常有好处。

异地机房的MySQL的重建问题

当某机房的所有Slave的数据出现问题（因为开发人员的误删除操作，所有的Slave都需要重建），或因为压力过大临近崩溃边缘需要紧急扩容，并且当前在线的任何一台机器都不能离线的情况下，就需要考虑利用备份来重建机房的数据库。

传统的做法是从有数据备份的机房把数据传到此机房，然后利用这个备份进行机房的重建工作，数据库文件小时倒是没有太大问题，如果数据库文件很大，跨机房传输数据就会大大降低恢复或扩容时间。对于某些对可用性要求比较高的业务来说，比较慢的恢复或扩容时间是没有办法承受的，这就需要在每个机房定期对数据进行备份，这个备份策略不同于日常的数据备份工作，每天备份一次，不用保留比较多的历史数据，当此机房的Slave集群出现问题时，使用本机房的备份来对本机房的Slave集群进行重建工作，这样不仅能提高重建速度，同时也节省多机房之间的网络带宽。

多机房部署中MySQL的监控问题

监控系统在MySQL的整个维护过程中起着非常重要的作用，决定着数据库可用性的高低。如果监控系统是在一个机房，那就会出现因为多机房之间的网络故障，而导致某机房的监控系统无法监控到其他机房的MySQL服务。这就要求有一套分布式监控体系，每个机房都有一套监控MySQL的系统，各个机房的监控系统都能独立工作，不会因为网络故障导致监控机检查不到MySQL服务器的状态信息。本地的监控系统监控本地的MySQL服务容易提高监控的频率，从而有利于提升整个机房的MySQL集群的可用性。

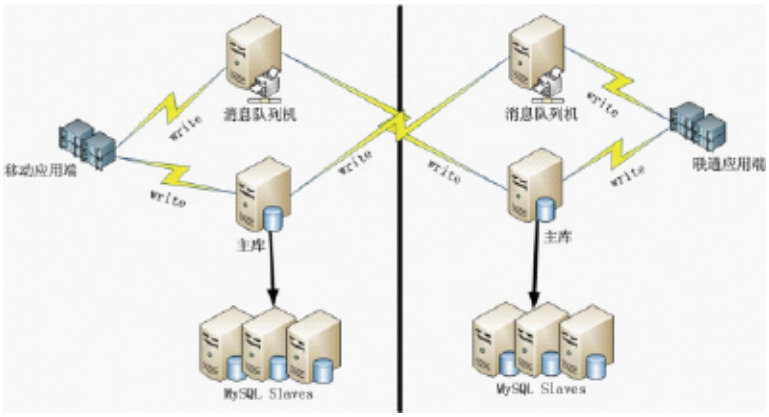


图2 利用消息队列实现两点写入结构

MySQL多机房设计中的可用性问题

之前也提到使用中继的方式进行数据同步，那么怎样保证这个中继的正常呢？中继作为另一个机房的Slave，同时作为本机房的Master，在出现故障时是不能很快恢复的，可以通过使用两台中继，一台机器上挂一半的Slave集群。在某一台中继出现问题时，不再对其下的 Slave进行访问，只访问正常中继下面所挂载的Slave机器，但前提条件就是一半量的Slave集群能承担住本机房整个业务的压力。使用Heartbeat+DRBD+blackhole引擎来提高中继机器的可用性，或者想好切换策略，在出现故障时切换到本机房到其他机房中速度最快的一个机房的Slave集群上，服务访问速度慢总比不可用要好。图1是一个多机房部署结构图，同时考虑了可用性问题。

多点写入问题

通过部署Master/Master（双向复制）、环状复制结构、树状复制结构（叶子节点之前的都可以写入），或利用Federated引擎都可以实现多点写需求。

多点写入需要考虑几个问题，一个是自增的问题，另一个是更新丢失的问题。自增问题比较容易解决，通过设置MySQL的两个参数来保证，例如有N个节点需要同时写入，那么所有的节点设置auto_increment_increment=N，每个节点依次设置为：auto_increment_offset=1,2,3,...,N，就可以解决自增问题。更新丢失的问题不是很容易解决，需要配合应用来做处理，例如：利用类似Views的概念，每个地方只更新本地数据，但所有Slave中的数据都是全数据。

用MySQL的复制结构部署的多点写入结构只适合于对次序要求不高的业务，如果业务层对写入次序要求比较高，每个机房写入的数据最终要以一定的次序表现出来，那么就需要配合应用来解决。例如利用发号器和时间结合来保证各个机房写入的数据在数据库里的次序最终一致。还有一种方式就是通过消息队列的方式，每个机房都有本地的主库和本地的消息队列，数据写入本地主库的同时，也写入到本地的消息队列中，本地的消息队列更新其他机房的主库，最终所有机房的数据都是一样的，是全数据。这样做有一个好处：避免网络线路故障对数据的影响，机房之间网络出现故障时，数据会暂时堆积在消息队列，等网络恢复后继续把堆积的数据分发到其他机房的数据库。每个业务都有高峰期，通过消息队列能使写入的波峰变成一条直线，硬件资源能准备到这条直线位置就可以，对资源的合理利用有一定的好处。但这种方式的前提条件是需要程序解耦。

图2是两个机房两点写入的部署结构，与多机房部署结构类似，每个机房的消息队列会向其他机房发送更新主库的消息，这个方案适合在三个以内的机房，当写入点增多时，消息的分发关系将成为一个网状结构，通信网络和各机房的主库将会成为整个系统的瓶颈，应用也会变得比较复杂，难以维护。

MySQL服务多机房部署中，都会或多或少地遇到这六个问题之一。如何部署一套满足业务需求的结构，需要根据实际情况以及业务特点进行设计。每种结构优缺点并存，唯有扬长避短，选择适合业务需求的结构才是优秀的部署结构。愿这篇文章能让你在MySQL的多机房部署中少走弯路，搭建出一套适合自己业务特点的系统架构。

作者简介 杨海朝



新浪首席DBA，在大规模高并发、海量访问特别是大规模数据库运维方面有丰富的管理和维护经验。热衷于数据库设计、性能优化、分布式部署方案和高可用性方面的研究。

责任编辑：董世晓（dongsx@csdn.net）

基于反射的软件自动化测试框架设计(下)

■ 文 / 黄捷

文章以Windows SDK工具箱中Service Configuration Editor的实际测试为例，演示了反射在测试中的应用。

在本文的上篇（请见2010年11月刊）中，我们讨论了测试框架设计的前两步：测试用例文件的定义和读取、用反射从相关程序集中获取类型信息。现在我们来继续探讨接下来的步骤：构造测试数据、执行用户界面（以下简称UI）操作、检查UI操作的结果。

基于反射的服务配置编辑器测试框架设计

构造测试数据

在用反射得到了测试目标背后的类型信息后，我们就可以根据这些信息构造测试数据了。测试程序最好提供这样的灵活性，让测试用例撰写者可以在测试定义文件中自己指定测试数据，也可以让测试程序随机生成数据。使用指定数据的优点是，可以保证某些特定的边缘情况在每一次测试中都被测到；而使用在一定范围内随机选取的数据可以在不增加运行时间的情况下增加测试的覆盖率，并省去了定义数据的麻烦。服务配置编辑器的测试以随机数据为主，在特定的情况下（如边缘情况测试和一些软件缺陷的回归测试）使用预定义数据。

在上一步中，我们已经得到了测试目标的类型信息和可配置属性信息。为了获得合适的随机数据，需要为每一个测试中用到的数据类型（如int类型）定义一个随机数据的列表，以供程序从中随机选取。我们可以根据需要将有效数据和无效数据分放在不同的列表中，分别用做常规功能测试和错误处理测试。在有效数据的列表中，我们应该尽量去包含所有可能的边缘情况（比如对于int类型来说，0，-1，最大和最小可能的int值等）。为每一个类型定义数据列表听起来是一件很费时的工作，但其实一般测试都用不到很多数据类型，而且有些类型的数据并不需要我们自己定义。下面的代码演示了如何用反射得到一个Enum类型的所有可能值，并从中随机选取：

```
// 新建随机数生成类的实例。请重用同一个实例而不要
每次新建，否则结果也许不是你要的随机结果
static Random dice = new Random();
// 得到所有可能Enum值
Array enums = Enum.GetValues(type);
// 随即从中选取一个并返回
return enums.GetValue(dice.Next(enums.
Length));
```

我们应该在测试运行之前就生成好所有的测试数据，而不是在测试进行的时候碰到一个

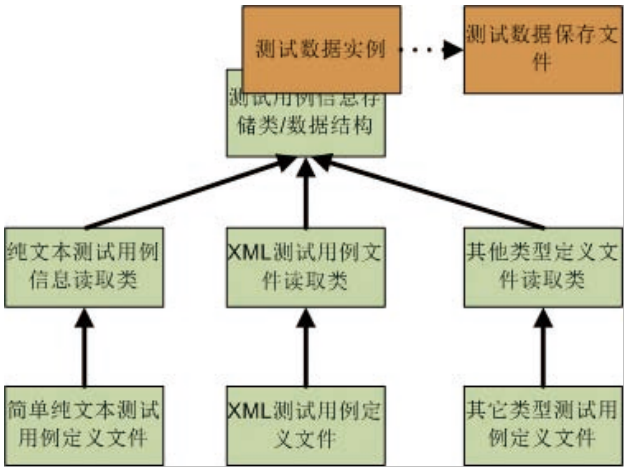


图5 测试数据生成和保存

生成一个。这样做的好处是，你可以很容易地把所有的测试数据保存下来，也可以很容易地用其他方式提供测试数据而不改变测试运行的代码（又一次接口与数据分离）。下面是测试程序在一次运行中为BasicHttpBindingElement随机生成的测试数据：

```
//仍然是上文提到的BasicHttpBindingElement
  的例子
BasicHttpBindingElement=>{
  Properties=>[
    AllowCookies => True,
      HostNameComparisonMode => Exact,
      MaxBufferSize => abcde,
      OpenTimeout  =>  10675199.02:48:05.
        47758071,
      // …… 省略此处部分内容
  ],
  ChildElements=>[
    ReaderQuotas(XmlDictionaryReaderQuotas
      Element)=>{
      Properties=>[
        MaxStringContentLength => 1.2345678,
        MaxArrayLength => 214748364721474
          83648,
        MaxBytesPerRead => IÇMESUYU SEBEKE
          INSAATI,
        // …… 省略此处部分内容
      ],
     ChildElements=>[
     ],
    },
    Security(BasicHttpSecurityElement)=>{
      Properties=>[
         Mode  =>  TransportWithMessage
        Credential,
      ],
    ChildElements=>[
    // …… 省略此处内容
    ],
  },
],},
```

这些数据都被保存在一个数据类型的实例中。通过这个实例，我们可以对将要运行的UI测试所使用的数据有一个很清楚的了解。另外，也可以很容易地把这个实例串行化并保存到文件中，在程序出现问题的时候从这个文件读入测试数据（简单反串行化就可以了）以重现问题。当然，你可以选择只在程序出现问题的时候才去保存这样一份数据文件。

需要注意的是，生成随机数据时，我们还需要正确处理数据间的依赖关系（有时候，一个输入数据的值取决于另一个数据值的选取）。我们需要另外的逻辑或者数据结构来维护这些依赖关系。这里不再赘述。

做完这一步，我们已经可以享受到反射带来的好处了。不管底层框架的开发人员对我们要测试的目标类型做怎样的调整和修改（比如给上面的BasicHttpBindingElement增加一个子元素或重命名一些属性），我们的测试代码都不需要做任何改动。非常酷，不是吗？

执行UI操作

首先，我们最好有一个比较通用的方法去找到并打开目标UI。对于服务配置编辑器来说，例如前文中提到的basicHttpBinding节点，如果我们能找到它在界面上节点树中所处的位置，就能打开它的属性配置界面。 而它在节点树中的位置又与它的对应类在底层框架中所处的位置相对应（可以想象，底层框架中也应该有一棵类似的树来表示所有的配置信息）。这个底层框架中的位置信息可以通过反射很容易地得到。所以，我们不需要为每个UI在测试文件里指明，或是在测试代码中硬编码它的位置信息，而是通过反射动态获取。当然，具体使用什么方法定位UI需要根据不同项目的实际情况而定。

接下来，我们需要一个操作目标UI的方法库。面向对象的UI测试会为每个主要UI定义一个抽象类型，对目标UI进行操作仅仅是调用它们的方法而已。测试到这一步要做的是根据上面生成的包含测试数据的实例，按步模拟执行用户的UI操作。这些操作的选择和步骤都在测试定义文件中定义。一个典型的服务配置编辑器的测试用例会做如下一些操作（还以

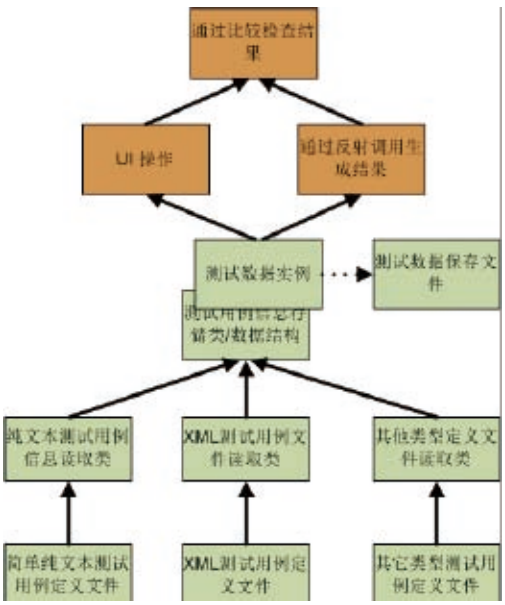


图6 UI操作结果的检查

basicHttpBinding节点为例）：

- 打开服务配置编辑器
- 新建一个配置文件
- 新建一个basicHttpBinding节点
- 编辑新建的basicHttpBinding节点
- 保存，关闭配置文件并检查结果
- ……

这样的操作被设定为默认的测试操作。如果只需要进行默认操作，在测试定义文件中只给出测试对象的名字（BasicHttpBinding-Element）就可以了。

注意：basicHttpBinding是这个绑定的显示名，BasicHttpBindingElement是它对应的内部类型名。

检查UI操作的结果

UI操作的正确性应该在操作的同时进行检查（比如通过检查UI的显示状态是否符合预期）。如果UI操作还有其他形式的输出，你需要另外对这些输出的正确性进行检查。这样的检查虽然因不同的测试项目而异，但有时也会有一些共同的东西。

在任何可能的时候，你都可以考虑一下反射是否能够继续发挥作用。如果你的UI像服务配置编辑器一样会输出一个文件，为了检查这个文件的正确性，传统的做法是事先定义好一个标准文件，在每次测试结束时，将输出结果与这个标准文件进行比较。这种做法的问题在于，每次产品代码更新时，我们都需要同步更新这个标准文件。

为了消除这个令人讨厌的维护工作，在一些情况下，可以用反射去调用UI操作背后真正调用的方法（注意，反射可以调用私有类型和方法），以此生成另一份输出，与UI操作的输出进行比较（前提是你无需保证被反射调用的方法的正确性）。

对于服务配置编辑器来说，因为UI与底层的编程接口是分离的，因此测试程序可以很容易地模仿UI对底层接口的调用，从而生成另外一份配置文件与UI输出的配置文件进行比较。而底层接口的正确性，并不在服务配置编辑器测试这一层来保证。图6是加上UI操作和结果检查两步后的设计图示。

总之，反射并不能够应用在所有的结果检查中。但是，在你准备写标准输出文件前，请先想一想应用反射（或者其他更好的方法）的可能性。如果可以应用，维护成本将会大大降低。

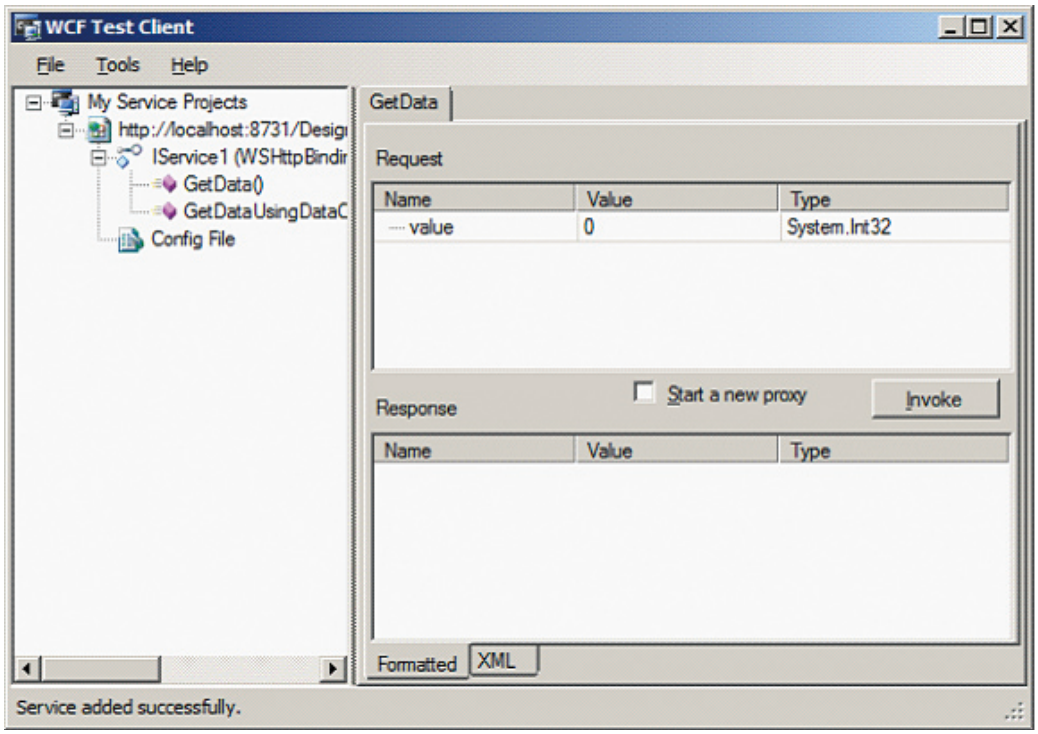


图7　WCF测试客户端

更多的反射

其实对于服务配置编辑器的测试来说，上篇图1中左边节点树中所有可能出现的节点（与配置文件中的元素相对应）都可以通过对根节点元素的反射来得到（它们都是根节点元素的后代节点）。所以我们不必为每个节点的测试编写测试定义文件，而可以通过反射来逐一测试所有的节点。当然，这样的做法还有很多问题要考虑，这也只是服务配置编辑器测试的一个特殊情况。这里不做过多描述。

再看一个与WCF有关的WCF测试客户端的例子。如果你在Visual Studio（2005及以后的版本）里面创建一个WCF Service Library项目（在新建项目窗口的WCF节点下面），按下F5，你就能看到它。图7是它的界面。

WCF测试客户端是一个用来测试WCF服务的客户端工具，用来帮助调试WCF服务，确保服务能被客户端正确调用。如果我们要开发它的测试框架，依据传统方法，我们需要预先写好各种不同种类的WCF服务。然后用WCF 测试客户端来调用这些服务，并查看这些服务返回的结果是否和预期一致。这些服务名称、地址接受的参数等信息都需要在测试定义文件或者代码中事先以某种方式给出，预期的返回结果也需要以某种方式告诉测试代码，甚至硬编码在代码中。但是，如果有了反射的帮助，不管给测试程序什么样的WCF服务，测试程序都能很容易地知道这些服务中提供了哪些可供调用的服务，以及这些服务接受了什么样的参数，返回了什么样的类型。有了这些信息，我们就可以像测试服务配置编辑器一样去动态构造这些参数，并通过WCF测试客户端调用这些服务操作。预期的返回值可以通过用反射动态构造WCF客户端并调用服务得到。将预期返回值同UI操作的返回值进行比较，就可以判断WCF测试客户端返回结果的正确性了。

举这个例子是为了说明基于反射的测试框架绝不仅适用于服务配置编辑器的测试。现在，不妨试试为一个你熟悉的图形界面的小软件做一下基于反射的测试框架设计的尝试吧！

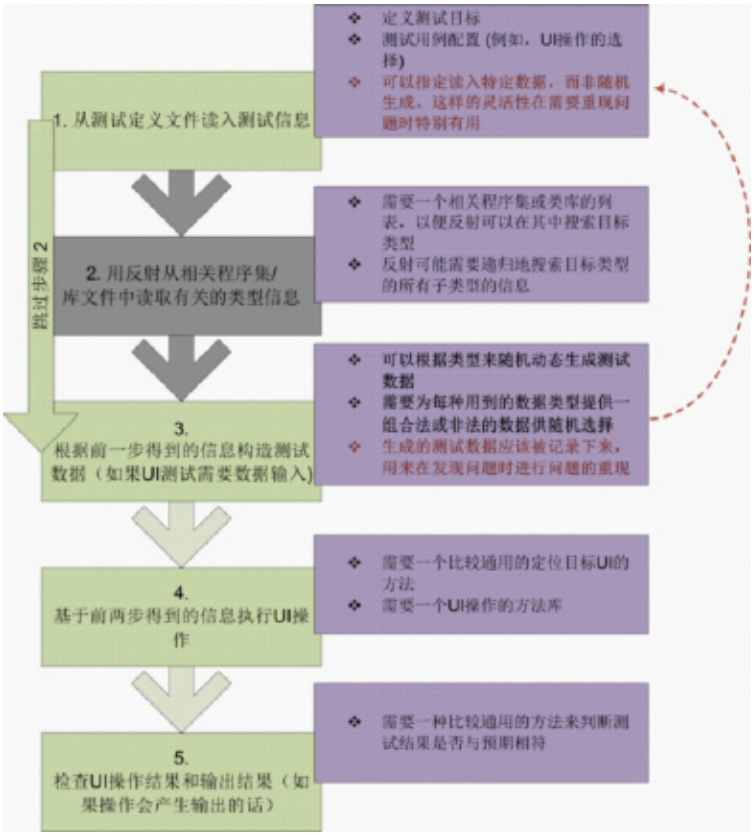


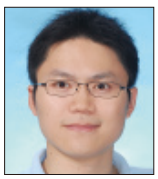
图8 出错场景的重现

总结

让我们再回过头来看一下整个流程。图8帮助你回顾每一个关键步骤中的一些要点。如上文中提到，当有错误发生时，我们可以把导致执行失败的随机测试数据保存到文件中去。随后通过修改测试定义文件，指定下一次运行从这个文件读入测试数据，以重现问题发生的场景。这个数据文件更可以直接被用作以后的缺陷回归测试（Regression测试），用来防止同样的问题再次出现。

最后，反射并不能应用到所有的测试项目中。但在设计测试框架时，应该留意运用反射以提高测试效率、降低测试成本的可能性。另外，虽然本文以C#和Windows平台为例，但反射的使用并不局限于平台和语言。希望本文能为实际测试工作带来一些帮助和启发。

作者简介 黄捷



2008年至2010年8月担任微软亚太研发集团服务器与开发工具事业部（中国）软件开发测试工程师，参与开发Visual Studio 及.NET Framework 相关工具。

责任编辑：董世晓（dongsx@csdn.net）

云计算的数据安全

■ 文 / 李季

本文从数据隐私和数据隔离两个技术角度，对云计算中的数据安全进行了阐释和剖析。

与传统的软件架构相比，云计算在运营和支持方面的成本更低廉，同时又能够获得更快速的部署能力和近乎无限的伸缩性等收益。尽管如此，仍然有诸多企业在云计算和传统软件架构中选择了后者，原因很大程度上在于云计算中，有关企业数据的安全问题没有得到妥善解决。一些分析机构的调查结果显示，数据安全问题是企业应用迁移到云计算的最大障碍之一。

数据安全意指通过一些技术或者非技术的方式来保证数据的访问是受到合理控制，并保证数据不被人为或者意外的损坏而泄露或更改。从非技术角度上来看，可以通过法律或者一些规章制度来保证数据的安全性；从技术的角度上来看，可以通过防火墙、入侵检测、安全配置、数据加密、访问认证、权限控制、数据备份等手段来保证数据的安全性。由于传统软件和云计算在技术架构上有着非常明显的差异，这就需要我们用不同的思路来思考两种架构下有关数据安全的解决方案。下面我从技术角度探讨一下云计算的数据安全。

数据隐私

对于任何一个IT系统来说，在运行生命周期过程中使用的和生产的数据都是整个系统的核心部分，而我们一般把这些系统数据分为公有数据和私有数据两种类型。

公有数据代表着可以从公共资源获得的数据信息，例如股票信息、公开的财务信息等，这类数据可以被任何一个IT系统获得并使用。而私有数据则代表这些数据是被IT系统所独占并无法和其他IT系统所共享的信息。对于公有数据，使用它们的IT系统并不需要处理安全相关的事务，然而对于私有数据特别是一些较为敏感的私有数据，在构建IT系统时需要专门考虑如何保证数据不被盗用甚至修改。

传统的IT系统通常搭建在客户自身的数据中心内，数据中心的内部防火墙保证了系统数据的安全性。和传统软件相比，云计算在数据方面的最大不同便是所有的数据将由第三方而非第一方来负责维护，并且由于云计算架构的特点，这些数据可能被存储在非常分散的地方，并且都按照明文的方式进行存储。尽管防火墙能够对恶意的外来攻击提供一定程度的保护，但这种架构使得一些关键性的数据可能被泄露，无论是偶然还是恶意。（例如，由于开发和维护的需要，软件提供商的员工一般都能够访问存储在云平台上的数据，一旦这些员工信息被非法获得，黑客便可以在万维网上访问部署在云平台上的程序或者得到关键性的数据。）这对于对安全性有较高要求的企业应用来说是完全不能接受的。

尽管目前在公有云平台还没有很好的数据隐私解决方案，但是企业仍然可以选择构建私有云或者混合云来实现弹性计算和数据隐私的均衡，同时也为未来在公有云平台上的实施积累经验。

从弹性计算的角度来看，私有云和公有云并无太大差别，都是通过自动化地管理虚拟化的IT资源来实现弹性计算。然而，由于现有的企业应用基本上都基于传统的IT基础架构搭建，几乎所有的IT资源都要求IT工程师花费大量的时间和精力来手动管理，并没有其他办法实现敏捷高效的自动化管理，因此无法满足云计算的要求。实施私有云计算的第一步也是最重要的一步便是重新搭建IT基础架构——将现有的处理器、存储、网络等IT资源高度虚拟化并重新组织整合，构建高度扩展性的IT集群架构，辅以强大的管理软件来实现高效自动化的IT资源管理。整个IT架构可以

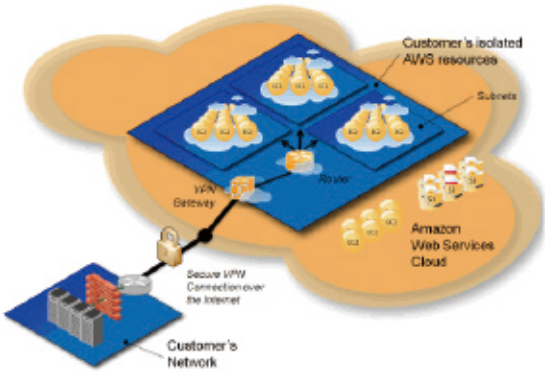


图1 Amazon提供的私有云解决方案

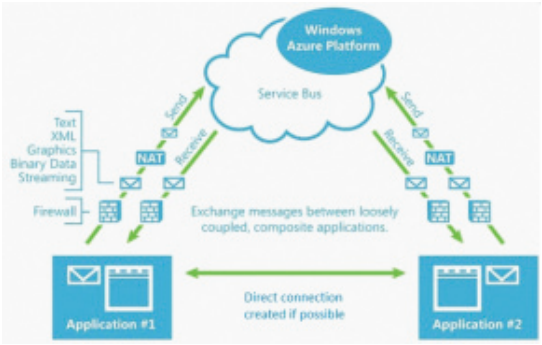


图2 Windows Azure AppFabric的混合云系统

搭建在企业内或者是第三方的数据中心内，但是无论私有云部署在什么地理位置，企业都拥有完全的IT资源控制能力。通过网络控制和独享的防火墙保护，私有云上的企业数据能够得到和传统IT架构下企业数据相同级别的安全保障。

在主流的私有云架构之外，Amazon的Virtual Private Cloud（虚拟私有云，以下简称VPC）提供了一套全新的企业级私有云构建方案。主流的私有云解决方案都致力于IT资源的虚拟化以及自动化管理，而VPC则将重点放在了如何构建专门针对单个企业的虚拟网络，并与企业现有的IT架构安全无缝地连接起来。企业可以在Amazon公有云平台上创建VPC虚拟网络，通过企业自身的加密VPN将VPC虚拟网络与企业局域网络连接起来，并将整个VPC虚拟网络加入到企业现有的安全架构下。在申请创建EC2实例时，可以将其指定与相应的VPC网络绑定，在EC2实例启动之后，该实例也就相当于运行在整个大的企业局域网之内（如图1所示）。虽然VPC网络中所有的EC2实例仍然位于公有云平台上，但是在这种IT架构下，企业内的防火墙能够保证这些公有云上的数据安全。

采用Amazon VPC的私有云解决方案无需对企业现有IT架构做出大规模的调整，因此也就无法减少IT运营成本，但是相比主流的私有云解决方案，实施成本和风险则减少了很多。

混合云则将云平台与In-House系统（自用系统）或者是私有云结合起来，将部分子系统搭建在企业内部的数据中心（通常这些子系统中的数据对安全性有非常严格的要求，或是一些Legacy系统（原有的系统）），而把系统的其他部分构建于云计算平台之上（通常这类子系统不带有数据安全性的问题），通过Service Bus（服务总线）将In-House系统（私有云）与公有云端系统连接起来。

图2展示了基于Windows Azure AppFabric的混合云系统：AppFabric提供了Service Bus的功能，私有In-House系统可以在AppFabric上创建WCF服务的Endpoint并注册，这样AppFabric便成为了公有云系统和私有In-House系统的中间平台，两个系统通过AppFabric来实现数据交互。由于Service Bus并不存在NAT（网络地址转换）的问题，因此并不要求企业局域网防火墙开放指定的端口，各个系统都能够比较方便地访问它。在最极端的情况下只需要服务器暴露80或者443端口，也就是服务能以HTTP或者HTTPS的方式连接上AppFabric。此外，AppFabric还提供了Access Control的功能，以支持常见的基于AD的身份认证。用户通过使用域账号登录，并将Token传给AC以获取相应的资源和服务。

数据隔离

IT软件服务商所提供的传统企业软件系统大多基于Multi-Instance（多实例）架构，即对于每一个客户组织，都有一个单独的软件系统实例为其服务；而搭建于云计算平台的软件系统则广泛采用了Multi-Tenancy（多租户）架构，即单个软件系统实例服务于多个客户组织。在Multi-Instance架构下，由于每个客户拥有自己的软件实例，所以不存在数据隔离问题，但是在Multi-Tenancy架构下，由于所有客户数据将被共同保存在唯一一个软件系统实例内，因此需要开发额外的数据隔离机制来保证各个客户之间的数据不可见性并提供相应的灾备方案。

随着云计算技术的成熟，Multi-Tenancy不

TenantID	CustName	Address
4	TenantID	ProductID	ProductNam
1	4	TenantID	Shipment	Date
6	1	4711	324965	2006-02-21
4	6	132	115468	2006-04-08
	4	680	654109	2006-03-27
		4711	324956	2006-02-23

图3　TenantID字段



图4　分离数据库架构示例图

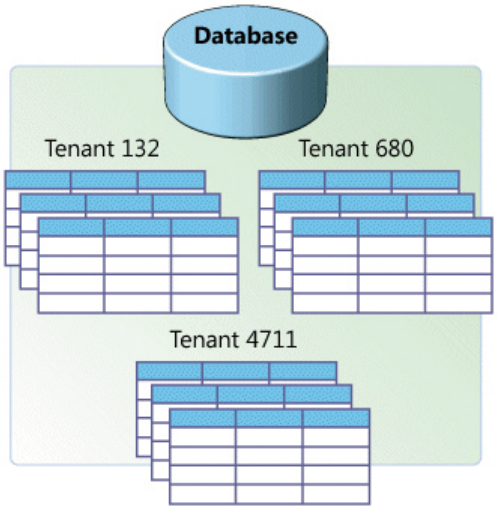


图5　分离表架构示例图

再是新鲜的概念，目前已经有几种成熟的架构来帮助系统实现数据隔离：Shared Schema Multi-Tenancy（下文简称为共享表架构）、Separated Database（下文简称为分离数据库架构）以及Shared Database Separated Schema（下文简称为分离表架构）。

- **共享表架构**：即所有的软件系统客户共享使用相同的数据库实例和相同的数据库表，但可以通过类似于图3中的TenantID字段来区分数据的从属。

因为共享表架构最大化地利用了单个数据库实例的存储能力，所以这种架构的硬件成本非常低廉，但对程序开发者来说，却增加了额外的复杂度。由于多个客户的数据共存于相同的数据库表内，因此需要额外的业务逻辑来隔离各个客户的数据。此外，这种架构实现灾难备份的成本也非常高，不但需要专门编写代码实现数据备份，而且在恢复数据时，需要对数据库表进行大量的删除和插入操作，一旦数据库表包含大量其他客户的数据，势必对系统性能和其他客户的体验带来巨大影响。

- **分离数据库架构**：即每个软件系统客户单独拥有自己的数据库实例，如图4所示。

相比于共享表架构，由于每个客户拥有单独的数据库实例，这种架构可以非常高效便捷地实现数据安全性和灾难备份，但是随之而来的缺点便是其硬件成本非常高昂。

- **分离表架构**：即软件系统客户共享相同的数据实例，但是每个客户单独拥有自己的由一系列数据库表组成的Schema。

分离表架构是一种折中的Multi-Tenancy方案，在这种架构下，实现数据分离和灾难备份相对共享表架构更加容易一些，另一方面，它的硬件成本也较分离数据库架构低。

无论是分离数据库还是分离表，抑或是共享表，每种架构都有它的优点和不足，在设计云端系统时，系统架构师需要进行全面的分析和考量，综合各方面的因素以选择合适的Multi-Tenancy架构。一般来说，系统服务的客户数量越多，则越适合使用共享表的架构；对数据隔离性和安全性要求越高，则越适合使用分离数据库的架构。在超大型的云系统中，一般都会采用复合型的Multi-Tenancy架构，以平衡系统成本和性能，这其中Salesforce.com便是一个典型的案例。Salesforce.com最初搭建于共享表架构，但是随着新客户的不断签入，单纯的共享表架构已经很难满足日益增长的性能要求，Salesforce逐步开始在不同的物理区域搭建分布式系统。在全局上，Salesforce.com以类似于分离数据库的架构运行，在单个区域内，系统则仍然按照共享表架构运行。

作者简介 | 李季



奥博杰天技术经理，主要致力于基于Amazon EC2以及Windows Azure平台的应用开发与研究工作。

■ 责任编辑：董世晓（dongsx@csdn.net）

移动游戏架构的艺术

■ 文 / 杨丰盛

游戏是一门综合的艺术，作者以铸剑为喻，形象地展示了移动游戏架构的艺术之美。

艺术是人的知识、情感、理想、意念综合心理活动的有机产物，是人们现实生活和精神世界的形象表现。文学可以是艺术，雕塑可以是艺术，音乐可以是艺术，电影可以是艺术……那么游戏呢？

游戏是一门综合的艺术

无论是单机游戏还是网络游戏、PC游戏还是手机游戏，都需经过如图1所示的逐步进化的过程。

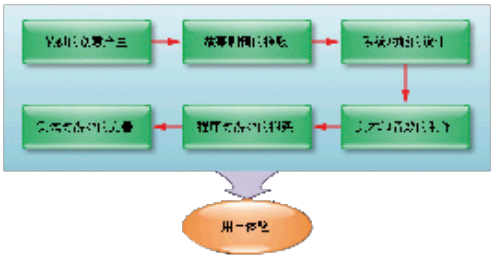


图1　游戏研发的流程

任何一款游戏，都有故事情节、人物、场景、音乐和音效等内容，因此其中每一部分内容都需要追求以艺术化的手段，将游戏的意象情景更生动地展现给用户。作为程序员的我们，之前可能只是以为只有策划人员需要通过艺术来表达游戏的内容、测试人员需要带着艺术去欣赏并完善游戏的体验，那么程序员对游戏的架构和艺术之间又是怎么挂上钩的呢？

用程序语言铸造艺术之剑

从古至今，有不少人热衷于铸剑，而铸出的剑是否称得上艺术作品，关键就在于它是否能表达出铸剑人的情感和用心所在。那么对于游戏程序的架构来说，也需要能体现出架构师对游戏的整体架构和每一个模块的把握，以及赋予它们的“神力”；同时，好的架构还将贯穿于游戏开发的整个过程。简单的测试一下，你是否也会遇到以下这些问题呢？

- 感觉游戏框架非常混乱，毫无条理？
- 开发完成后出现大量Bug，并且不知从何下手去解决？
- 出现很多难以重现的致命Bug？
- 添加某个功能或模块非常困难？
- 经过频繁的修改之后，出现大量冗余代码？

如果出现上述问题，那么说明我们所使用的这把架构之剑并不锋利，可能没有将“力”用在剑刃上吧。这也从另一方面体现出铸剑过程（也就是游戏的架构）的重要性。如果带着这些问题继续锤炼，那么终将铸成一把绝世宝剑。

架构是一个看不到也摸不着的虚幻东西，因此，我们还是结合具体的实例来分析如何将架构艺术化，通过代码来表达架构思想。假设现在有一套策划案，并且美术人员能够为我们准备所需要的资源。

“采矿”是铸剑之根本

有了铸剑的意念，就需要去选择各种高品质的玄铁矿石。同样，对于游戏的架构来说，就需要对游戏的各个功能模块进行分析，针对每个模块需要实现的功能和特殊属性进行抽象，概括出各个模块所需要定义的数据结构和接口以及各模块之间的关系。然后，再根据策划文档整理出适合编写程序的流程图、关系图、结构图等。最后，根据这些文档资料配合游戏的运行环境（比如游戏运行的平台）对实现技术进行需求分析，该分析主要针对功能和性能进行。下面我们以Android和iPhone两大平台为例进行分析。

在Android平台上，从效率的角度考虑，可以选择C/C++作为开发语言、OpenGL|ES作为图形库；从简单方便的角度考虑，可以选择Java作为开发语言、Skia作为图形库。

而在iPhone平台上，从跨平台的角度考虑，可以选择C/C++作为开发语言、OpenGL|ES

作为图形库；从开发的难易程度考虑，可以选择Objective-C作为开发语言、Quartz2D作为图形库。这里我们没有考虑3D的特殊问题，如果进行3D游戏开发，必然会选择OpenGL|ES。另外，在iPhone平台上使用Quartz2D作为图形库开发游戏的比较少。

如果要做的是一个跨平台的游戏，我们还需要分析各平台的跨平台能力和特性（包括开发语言、渲染库、资源管理、内存管理等），例如iPhone中内存管理采用了引用计数的机制，那么可以考虑通过C++来实现一个类似的机制，这样就可以保证其他平台也都可以通用该机制。

这里我们也没有刻意地对某个语言和库进行比较，因为它们通常都有各自的优势，需要根据具体的游戏需求进行选择。准备好铸剑的玄铁之后，下一步我们开始铸剑（编写代码）了吗？

“剑范”决定“剑气”

“剑范”即铸剑所用的模型，“剑气”即剑品的气势，因此，铸剑所用的模型将最终决定宝剑的“主题”。

“剑范”的打造对应于游戏架构中负责全局范围的核心框架，该框架是否合理，将决定整个架构的成败。说到框架，不得不说说大家可能都比较熟悉的MVC框架，作为程序员，应该没有谁不知道MVC的定义，但是关键在于如何从更深层次去学习它，理解它，运行它。曾经一位朋友问我：“你们通常采用什么框架？什么框架会比较好呢？”我回答：“MVC框架啊！”看着他一脸吃惊的表情和一句“也很普通的框架！”的回答，我真不知道说什么好！MVC框架的确很普通，也很简单，而如何让它发挥出其本身的功效却是关键！例如，在iPhone开发中，苹果公司就推荐采用该框架，具体描述如下：

- 视图（V）：由UIView类的子类及其相关的UIView-Controller类来提供；
- 控制器（C）：控制器行为通过主要由委托、目标操作、通知来实现；
- 模型（M）：通过数据库和数据含义等协议提供逻辑模型，由控制器进行管理。

另外，就连目前红遍整个互联网的Android系统来说，Google官方也同样推荐使用MVC框架来开发和维护应用程序，具体描述如下：

- 视图（V）：由View和SurfaceView（或者自定义的视图）提供用户的界面，采用XML格式进行布局；
- 控制器（C）：由Activity来管理视图和逻辑模型；
- 模型（M）：向视图提供逻辑模型，由控制器来触发，比如各种Adapter接口。

根据切身的经历，我得出了下面的观点：“如果不能保证将其他的设计模式运用好，就不如选择类似于MVC这类的简约而不简单的框架来完成，可能结果会更精彩！”

由此可见游戏架构中框架的重要性。如果你还觉得不够，那么我建议应该向众多的Android和iPhone开发者询问答案，苹果和Google的文档将是你最好的学习资料！相信你已经跟着我们的脚步，打造出了适合你自己的“剑范”了，现在是时候开始铸剑了！

真金不怕火炼

宝剑品质的好坏在于使用的原材料，如何检验矿石的材质？还得看准备的火力是否强劲。现在让我们回到游戏架构的话题。

首先，尽管通常游戏中可能会存在多个视图、多个模型甚至多个控制器，我们还是将一款游戏分为三个大的模块，即MVC的控制器、视图、逻辑模型。图2为我当前正在开发的一款Android游戏的框架。

由图2可以看出，程序从入口Activity进入，通过setContentView()函数设置显示一个视图类MainView时，游戏的主线程已经开始Loop了；而游戏要显示的具体场景则由控制器Control类根据游戏状态等信息通过getCurScene()函数得到，得到的场景实际上是抽象场景SceneInterface类的子类（如SceneA）；在SceneInterface中包括各个场景的公共接口，以便在Control.control()中能直接控制当前显示的场景，需要注意的是，在控制器中一般需要先通过SceneInterface.reCycle()来对

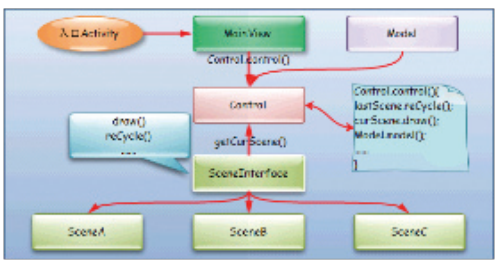


图2　MVC游戏框架

上一个界面产生的垃圾进行回收，同时对逻辑模型类Model中的数据进行处理。

由于图片大小的原因，我没有列出全部类结构，这些类包括在SceneInterface中定义一个界面事件的公共接口或者对Model采用“实体—对象—类”的方式进行扩展以及所需的工具类。

铸剑的最高境界——指导练剑之人

好的宝剑通常可以提高用剑之人的剑术！Google和苹果所铸造的Android、iPhone这两把宝剑，指导了无数的开发者。我们又何不学习学习Google和苹果这两位铸剑人的技艺呢？

下面继续分析Android可能会如何指导“练剑人”。在做Android应用开发时，Google推荐通过XML文件来布局界面，从而让UI界面和程序逻辑分离，方便维护；实际上Google对XML文件以及其他资源都进行了优化，这样做可以提高程序的运行效率。下面我们也将学习这一招，通过XML文件来布局游戏场景。

首先，将一个游戏场景（SceneA）中需要显示的内容分为UI界面（如按钮、菜单等）和游戏对象（如精灵）两部分；其次，UI界面和游戏对象可能都需要动画；最后，既然要使用XML文件来布局，就需要一个专门的解析器。稍作整理，现在我们所需要的模块如图3所示。

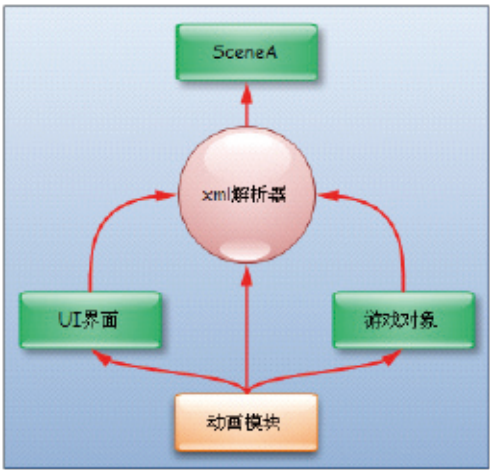


图3　游戏场景架构

图3描述了一个场景的元素架构与解析，其中UI界面、游戏对象、动画模块都将使用一个或多个XML文件来完成，通过XML的解析，最终将XML文件的内容输出到SceneA中。整个过程首先需要构建一个XML解析器，然后根据游戏的需要构建各种UI控件、游戏对象、动画等类。目前工程中用到的类包括：

- UI控件：UIText、UIProgress、UITexture、UIList、UIClick、UICheckBox
- 游戏对象：GameObj、EffectObj、Bullet、Enemy
- 游戏动画：PosAnimation、ColorAnimation、FrameAnimation

每一个类中都有其所需要的成员变量，下面我们可以根据解析器的实现来使用XML文件布局游戏场景了。根据游戏的需要，可以将XML文件扩展到支持多级子节点，然后在游戏中获取XML中定义的元素，进行操作即可。同样需要将界面和程序进行分析，方便修改和维护。整个过程的完整流程如图4所示。

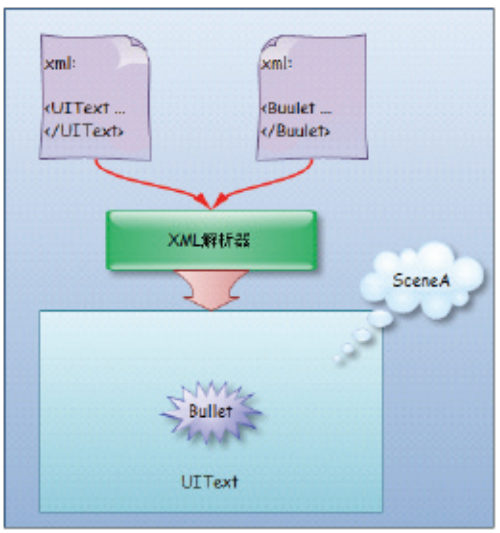


图4　XML布局流程

实际上我正在进行的项目除封装了上面所说的三个模块外，还将整个游戏架构都进行了封装，包括图2中所展示的控制器、主线程、模型，整个框架都能通过XML文件直接编写。到这里，你所铸造的宝剑也应该出世了，带上宝剑，冲向游戏世界吧。

作者简介 杨丰盛



轻灵数码研发部经理，精通Java、C/C++等语言及J2ME、BREW、MTK、Android、iPhone等平台。曾领导和参与《三国群英传》、《大航海传奇》等游戏的开发。著有《Android应用开发揭秘》。

■ 责任编辑：董世晓（dongsx@csdn.net）

趣谈CLR4的调试模型重构（上）

转眼间，2010年就要结束，当读者朋友读到这篇文章时，就是2011年了。原本计划要在2010年写一本书的，但是从年初到年底也没写出一章来，只是脑海中悬浮着一篇篇空想，真是惭愧。略微可以安慰一下的是，这个调试之剑专栏还坚持着，每月一篇，风雨无阻。有些同事见到我，会好奇地问，你怎么有那么多东西可以写。我一般会笑着回答："软件的问题太多了，还写不过来呢！"是啊，除了时常有活生生的问题冒出来外，我的硬盘上积攒了接近8GB的转储文件，有系统问题，也有应用程序问题，足够用作几百篇文章的素材。总体说来，2010年的文章，大多数是实战性的案例分析，比如系统挂死、应用程序死循环等。新的一年中，除了继续坚持高趣味性和挑战性的实战内容外，应某些读者的要求，我们会穿插安排一些调试原理的内容，帮助大家夯实理论基础。比如，这一期和下一期我们将围绕.NET 4（CLR4）中的调试模型重构向大家介绍有关的调试原理。因为我们侧重的是调试原理，不熟悉.NET的读者也不必担心看不懂，我们先从著名的海森伯效应说起。

海森伯效应

海森伯效应（Heisenberg Effect）来源于德国著名物理学家沃纳·海森伯（Werner Heisenberg）的测不准原理（Uncertainty Principle）。这个原理指出不可能同时精确地测量出粒子的动量和位置，因为测量时仪器会对被测量对象产生干扰，测量其动量就会改变其位置，反之亦然。测不准原理也被称为不确定原理，即测量的过程会影响被测试的对象。换句话来说，因为海森伯效应的存在，测量的过程会影响被测量对象使测量结果不准确。

在软件领域，人们把调试时无法复现或者行为发生改变的错误（Bug）称为海森伯错误（Heisen Bug）。因为在调试环境下无法稳定重现，所以调试海森伯错误通常更加难以解决。

为了降低调试过程对软件错误所造成的影响，设计调试器的一个最根本原则就是使海森伯效应最低。换句话来说，就是要使调试器对被调试对象的影响尽可能的小，或者说二者的关系最好是互不影响、互不干涉。但是为了实现调试功能，调试器又必须与被调试程序建立起比较密切的联系。也就是说，强大的调试功能要求调试器对被调试程序有较高的可控性，要求二者建立密切的关系。而海森伯效用又要求调试器和被调试程序的关系不能太紧密。看来这两者之间存在着一定的矛盾，如何平衡这个矛盾是设计调试模型的一个关键问题。

本地应用程序调试模型

有了海森伯效应的基础后，我们先来看一下Windows操作系统中调试本地应用程序（Native Applications）的调试模型。图1画出了从Windows XP开始一直使用至今（Windows 7）的调试模型原理图（XP之前略有不同），左上部分是被调试进程，右上方是调试器进程，下半部分是操作系统内核。

概括来说，Windows本地应用程序调试使用的是所谓的进程外调试模型（Out-of-process Debugging Model），调试器与被调试程序是两个独立的进程，二者通过位于内核中的调试对象和调试子系统（DbgK）建立稳定的调试关系。因为主要的调试设施都位于调试器进程和内核中，调试行为对被调试进程的影响非常



主持人：张银奎

《软件调试》一书作者，从事软件开发和研究10余年，对IA-32架构、操作系统内核、虚拟技术，尤其对软件调试有较深入的研究。翻译（合译）作品包括《数据挖掘原理》、《机器学习》、《人工智能：复杂问题求解的结构和策略》、《观止——微软创建NT和未来的夺命狂奔》等。

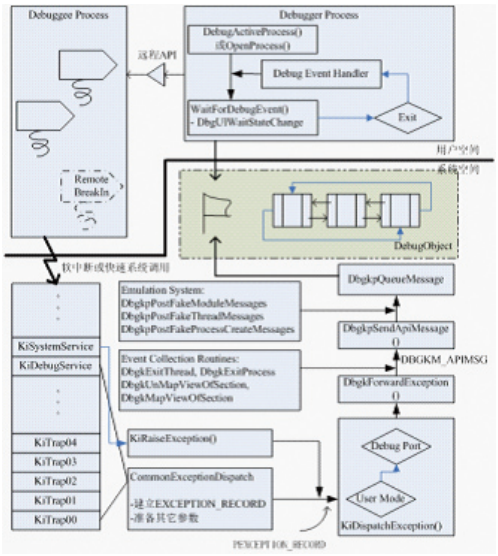


图1　Windows本地应用程序调试模型（XP及以上版本）

小，所以进程外模型的海森伯效应较小。多年来的实践证明，Windows的本地调试模型工作得非常好，如果探寻Windows平台上的应用程序如此繁荣的技术原因，那么这个优秀的调试模型功不可没。

CLR4之前的托管调试模型

下面再来看一下，.NET 4之前用于调试托管代码所使用的调试模型，因为.NET 4前使用的主要版本是.NET 2，所以我们简称CLR2托管调试模型。图2是重构前的托管调试模型的原理示意图，左侧是调试器进程，右侧是被调试进程，调试器进程与运行在被调试进程中的调试辅助线程（图2中右侧的原型箭头）通过进程间通信进行沟通，共同完成各种调试功能。

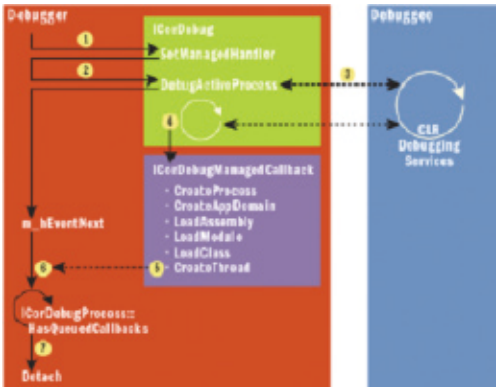


图2　重构前的调试模型

在重构前的托管调试模型中，调试辅助线程承担着核心角色，负责采集各种调试事件和执行访问调试目标的各种功能，从调试模型的角度来看，辅助调试线程是调试器的关键部分，因此与前面介绍的本地调试模型不同，重构前的托管调试模型属于进程内调试模型（In-process Debugging Model）。进程内模型的优点是调试器可以更便捷地访问调试目标（位于同一用户空间中），但也有明显的缺点，首先调试设施可能影响被调试目标，具有较高的海森伯效应，另外，调试设施也容易受到调试目标的影响而无法正常工作，下面将通过实例来说明这些局限性。

“二调一”

因为重构前的托管调试会话没有使用本地调试会话所使用的调试端口等设施，所以当托管调试在进行时，还可以使用一个本地调试器附加到同一个托管程序上，形成两个调试器同时调试一个程序的“二调一”现象。

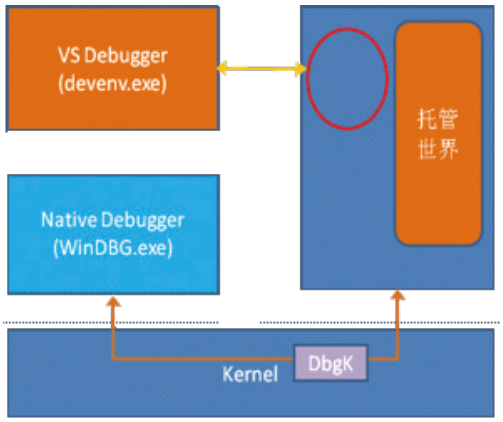


图3　二调一（重构前）

感兴趣的读者，可以亲手做个实验。运行一个.NET 4之前（比如.NET 2）的托管程序，先使用Visual Studio（2010版本，下同）以托管调试方式附加到这个程序，然后再运行WinDBG，也附加到这个托管程序。在WinDBG中，执行~*命令列出所有线程（图4），便可以看到调试辅助线程（即图4中的1号线程），它的入口函数为DebuggerRCThread::ThreadProcStatic。

为了感受辅助调试线程的作用，可以在

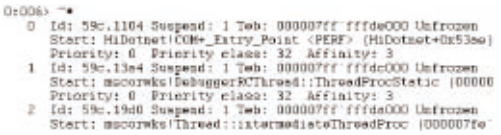


图4　观察辅助调试线程

WinDBG中执行~1n增加这个线程的挂起计数（Suspend Count），然后执行g命令，恢复被调试的托管程序运行，因为调试辅助线程被手工增加了挂起计数，所以当进程恢复运行时，这个线程会仍然保持挂起状态。而后，在Visual Studio中选择Debug菜单中的Break All…试图将托管程序中断下来，点击后，会发现Visual Studio程序陷入了“僵呆”状态。再运行一个WinDBG附加到Visual Studio进程（devenv.exe）上，执行~*观察各个线程的状态，会发现0号线程正在通过RPC方式与调试辅助线程通信（图5），因为调试辅助线程挂起了，所以这个线程陷入等待。



图5　Visual Studio与调试辅助线程通信的过程

“僵呆”几十秒钟后，因为RPC通信超时，Visual Studio才会苏醒过来，显示出图6所示的“致命错误”对话框。



图6　与调试辅助线程失去联络后的错误对话框

混乱的混合调试

看到这里，有些读者可能会问，为什么要使用两个调试器来调试同一个程序呢？我们这样做的目的有两个，一是用来演示调试辅助线程发生故障时对调试器的影响，二是模拟混合调试的情形。所谓混合调试，就是同时调试托管代码和本地代码。图7画出了混合调试的调试模型，图中的LS代表左端（Left Side），RS代表右端（Right Side），二者共同完成托管调试功能，RS上面的本地调试引擎用于提供本地调试功能，这其实相当于把上面试验中的两个调试器合并在一个进程里。Visual Studio的混合调试功能使用的便是这个模型。

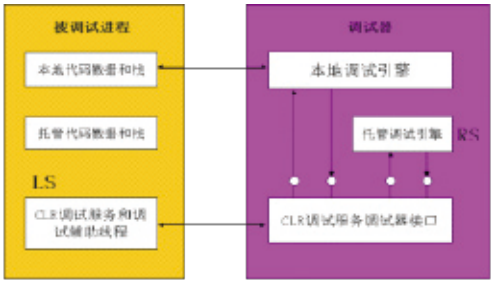


图7　混合调试示意图

混合调试的最大问题就是当本地调试事件发生时，进程被操作系统完全冻结，本地调试引擎得到调试事件后需要判断该事件是否属于托管调试引擎：如果属于，那么托管调试引擎必须先唤醒辅助线程才能开始工作；否则，就可能出现图6所示的托管调试器“僵呆”问题。因为这个原因，混合调试一直是.NET中难以解决的问题，直到今天，Visual Studio仍然不支持64位托管程序的混合调试。

转储分析中的难题

重构前模型的另一个大问题就是分析转储文件困难，在调试活动的目标时，调试器依靠调试辅助线程来访问托管世界，解析其中的语义。而当分析转储文件时，调试辅助线程不复存在，因此很难解析托管语义，常常无法得到完整的栈回溯，尤其是其中的托管函数只能看到JIT编译后的二进制函数地址，无法解析出对应的函数名称。以图8所示的栈回溯为例，某个托管函数抛出异常导致程序崩溃，但是却无法得到这个托管函数的名称。

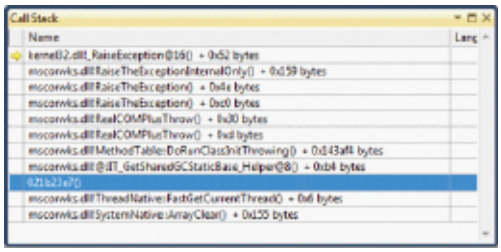


图8　转储分析的不完整栈回溯（重构前）

■ 责任编辑：董世晓（dongsx@csdn.net）

本期问题：

上期答案是：可以判断出关键区不是处于空闲状态，执行dt _RTL_CRITICAL_SECTION 433a288+4命令可以显示出这个关键区结构，LockCount字段的最低位为0以及OwningThread字段不为0（18e4）都表明这个关键区处于繁忙状态。

本期问题是：图8中突出显示的栈帧对应的函数返回地址021b23e70为什么无法解析到属于哪个模块？

编者说明：

- 投稿信箱：contest@csdn.net
- 联系方式写在一个单独的TXT文件里，包括以下几项：
  1）姓名
  2）工作单位或学校
  3）电话联系方式
  4）邮寄地址
  5）E-mail地址
- 解答提交时间，最好早于当月15日。

iPad上的设计工具

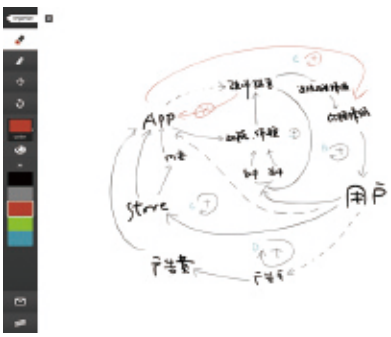
■ 文 / 苍浪

iPad可能是2010年最受欢迎的电子产品，作为前端设计师，本文作者教你如何在iPad上完成产品设计，让你同时收获工作成果与快乐。

美国《时代周刊》日前评选出“2010年十大电子数码产品”，苹果平板电脑iPad力压其他产品，位居榜首。2010年自iPad发售以来得到了各方青睐，以至于不管是什么促销活动，奖品如果不是iPad或者iPhone4都不好意思说自己有奖品。随着越来越多的人使用iPad，App Store的iPad应用也越来越多，社交、新闻、游戏、工具、导航等等，很多人成了应用的“收集控”，但大多数应用就打开用过一次就再也没用。我认为从如此多的应用中找到适合自己的，并能用于持续提升生产力、改进生产效率的应用才算得上好应用。经过几个月的不断试用，我整理了一些不舍得删除的应用，把这些应用按照产品设计不同阶段（需求采集、需求设计、需求展示）进行分类，接下来分别给你介绍适合思维导图、交互设计、需求展示的应用，另外还附带了一些同步工具，希望也可以对你有所帮助。

思维导图类: Adobe ideas、ibrainstorm、 Mindo

Adobe ideas是一款画图板，其实就是N张白纸，随时记录你的想法，这就是这个软件的全部功能。



Adobe ideas

（本图版权归白鸦http://uicom.net所有）

有时候你想表达一个观点，可能会用十多页PPT还说不清楚，而有时候你会发现几个简单的循环图、象限图或者几条曲线就可以把你原本想要用十多页PPT还不一定能表达出来的东西说明白。也许你就是在某个会议开小差，在iPad上随手画的画就为你理清了接下来



ibrainstorm



Mind

几个月的工作重点甚至全年的战略，而这个软件就是做这个的，虽然功能不多、很小，但足可以满足你的需求，让你不要遗漏瞬间的灵感。

ibrainstorm这款应用看名字就知道是做头脑风暴用的。在项目需求收集阶段，经常会有头脑风暴，在头脑风暴的时候，我们会拿很多张便笺，各自天马行空地提出各种想法，填满整个白板。而这个应用就是把线下的头脑风暴的场景放在了iPad上，随意添加便笺，在便笺上写上你的想法，在白板上整理构思，最后去掉无用的信息，整合需要的信息，得出结论，非常短、平、快的做法。也许你在家里，倚在靠枕上也可以来一次风暴！

而Mindo就是一款实实在在的画思维导图的应用。在PC上我们用的最



iMockups

多的是Mindmanager，而在iPad上画脑图最好的就是这个工具了。它不仅可以如Mindmanager一样在各个子项中添加进度图、优先级标识等，还可以导出MMAP文档，也就是说可以和Mindmanager通用，在iPad上没有完成的图，你可以继续在电脑上完成。不过你得有Dropbox账号。图画完之后，会让你导入到Dropbox，Dropbox直接在你的根目录创建一个Mindo的文件夹存放这些图。

交互设计类: iMockups、Sketchy、OmniGraffle

交互设计师们几乎天天要画各种原型图，给业务方展示自己的产品设计，iMockups是一款在iPad上的手绘风格的原型图绘制应用工具。

和PC端上的Balsamiq Mockups类似，它可以让你在产品设计的需求阶段用手绘的方式迅速记录你的灵感，结合iPad这个移动终端，让你的界面设计在不经意间就可以保留。主界面的左侧是树状导航，意味着你可以在一个项目下建立多个页面，右侧是组件库。为了便于设计在iPad、iPhone上的应用，iMockups 默认包含了iPad、iPhone组件，同时在你进行页面布局的时候会自动帮你对齐，这点是借鉴了Mac上Keynote的效果，非常好用。不过不足之处在于只能导出PNG图片，也不能和Axure通用，当然作为手绘草图已经足用。

除iMockups之外，Sketchy也是一款原型设计图，它的组件风格更接近于Axure，并且可以导出适合在Balsamiq Mockups上使用的bmml文档。而OmniGraffle的功能也很强大，除用来画思维导图、流程图、组织结构图以外，也可以用来画原型图，但画的时候稍显麻烦。

需求展示类：百家

也许你在iPad上订阅过周末画报或者牛博，在iPad上看杂志是件很舒服的事情，画面精美流畅，操作感强。你可能也想做一个杂志，有可能是公司、个人展示，有可能是一些相册记录，百家就是这样一个应用。一个iPad电子杂志的阅读和自助出版解决方案。你可以通过“百家”App来订阅别人出版的电子杂志，也可以通过“百家”自助出版平台来发布和管理自己的电子杂志，让别人通过“百家”App来订阅。



百家

要想在百家上发布杂志非常容易，你可以直接将PDF软件转化为在iPad上看的杂志，目前还不能添加声音、视频，估计接下来就会发布。试想如果可以直接导入Flickr、Yupoo甚至QQ相册里面的照片，再直接制作成电子相册，然后通过蓝牙或者wifi传给朋友，是一件多酷的事情。另外，现在很多人都是通过博客来传播自己的思想，也很想把自己写的东西出版。通过百家就降低了出版的门槛，使更多的人可以用iPad来阅读自己制作的杂志。如果杂志很有价值，百家完全可以提供小额支付渠道，作者和百家分润，在未来会有很大的发展前景。

GTD+同步工具：GoodReader、Evernote、OmniFocus

Goodreader是一款加强版的阅读工具，很多人是因为找Google Reader发现的它，但它们有本质的不同。

你上传文件至iPad，然后就可以直接读取。它支持读取的文件格式也很多，你不需要再下载Office的APP，可以直接查看Word、Excel、PPT和PDF文档。不过不好的是，查看PPT只能是文档的浏览效果，不能演示PPT的动画、翻页。另外很强的是，它可以直接读取你存在Dropbox（不用修改hosts）、Google Docs等的文件并下载，甚至还可以连接你的FTP服务器，这些对于同步控来说简直太棒了。如果再配合上Chrome或者Firefox的Evernote插件做文章手机、OmniFocus做时间管理，我想就可以让你在工作、学习中的效率提升一个层级。



Goodreader

最后还要说的是，工具只能让你更快速、更高效地工作，而比工具更重要的是坚持，如果可以坚持，什么工具都可以变为神兵利器。

作者简介 | 苍浪



苍浪，就职于支付宝（中国）网络技术有限公司。平时关注用户体验，关注客户感受，长期混迹于互联网各个角落，同时是Twitter和微博的重度用户。

■ 责任编辑：高松（gaosong@csdn.net）

主流RIA技术分析与对比

■ 文 / 覃旋

随着Web应用与需求的发展，RIA（富互联网应用）以其表现力强、用户体验良好的特点，受到广大技术人的热捧，本文分析比对了三种流行技术Sliverlight、Flash、HTML5，孰优孰劣，文中见分晓。

未来，“用户体验”将成为所有软件商业价值的首要衡量标准。拥有极好用户体验的RIA技术近些年来发展迅猛，其中以Silverlight、Flash及HTML5最受热捧。互联网上98%的计算机都有安装Flash；HTML5的新特性则强化了Web网页的表现力；Silverlight更是携带着高性能与高生产力等诸多优势异军突起。RIA技术领域逐渐显露三雄割据之势。

Flash发展至今已有15年历史，时间长河使之在全球孕育了大规模热衷于网页矢量动画的设计者与开发者，以及让无数商家垂涎的插件装机量。在Macromedia和 Adobe的号召下，无数优秀的艺术家、UI设计师和互动程序设计师团结一起，最终成就了今天全盛时期的Flash。

ActionScript是Flash最主要的编程语言，作为当下主流的RIA技术，Flash十数年的积累使其拥有极深的根基，Flash的主导地位短时间内很难被撼动。然而，所有问题都是相对的，时间的沉淀伴随而来的是漏洞百出，Flash的滥用和参差不齐的Web前端开发人员造成了当前很多网页中的Flash应用效率低下、浏览器极易崩溃等问题；插件越来越臃肿却依旧无法改变封闭的SWF对搜索引擎的不友好支持。更让诸多Flash Fans伤心的是，苹果方面明确地排斥Flash，从iOS到2010 MacBook Air都没有Flash的身影，或许Flash真已渐入中年。

HTML是目前网络上应用最为广泛的语言，也是构成网页文档的主要语言。HTML4标准已经十多年了，随着世界IT技术的飞速发展，各类软件和平台的不一致性骤增，至今我们还得继续兼容IE6下的CSS样式名，开发差异化的产品，却要提供同质化的功能，!important、*+、IE Only依旧是无数网页设计者心中的痛。HTML 4在各类平台上所受到的诸多限制使得HTML5呼之欲出。于是，2004年HTML5第一次被WHATWG提出，2007年最终为W3C所接纳。内置Canvas、Audio/Video以及Web存储等诸多新特性的HMTL5未来将会引领互联网多方面的改进与革新，依旧利用HTML + JavaScript + CSS使得普通网站也能轻松具备更丰富的功能，被世界所公认真正做到跨平台的技术。

然而，看似美好的HTML5却注定一生坎坷，从它出生开始就已被Google所挟持，企图号令天下。Google的云计算帝国就差一个RIA的支持，试想 HTML5如能普及，更多的应用转向B/S模式，微软帝国的桌面软件生态环境必定遭受巨大威胁，虎视眈眈多年的Google将是最大受益者，Chrome OS界面是最有力的证据。

作为Google的一个棋子，HTML5终究无法取代Silverlight或Flash等网页插件技术，因为网页插件能实现HTML5实现不了的功能，且拥有比HTML5更灵活的框架，整体性能更好、开发效率更高并具备跨浏览器(不同浏览器下无差异体验)等优势。另外，HTML5若要真正应用到商业领域，前提是其规范必须得到统一。可现状令人堪忧，微软的IE、Mozilla的Firefox以及主导者Google的Chrome当下都在极力地争夺HTML5标准制定方面的最强话语权，利益驱使下的HTML5标准统一进程5—10年内依旧一片



浮云，对于Silverlight和Flash来说这么漫长的时间已足够开发出几代新版本了，同时亦印证了W3C对当前HTML5/CSS3标准现状的表述：“不适宜用作生产环境”。

新秀Silverlight是微软所发展的Web前端应用程序开发解决方案，2007年9月这个奇迹般的小子横空出世，给本已混乱的Web市场再添一大变数。Silverlight最大的特点在于它精简的.NET框架，你可以使用C#、VB、F#这3大.NET主流开发语言去编写Silverlight应用程序。Silverlight的开发者大多来自传统的.NET产品研发一线，拥有国际顶尖的架构经验和丰富的设计模式知识，在VisualStudio以及Blend等强大开发工具的协同支持下，高性能、高开发效率等诸多优势显露无遗。Silverlight的应用主要集中在电子商务、视频、网页网络游戏、卫星定位、导航、地理系统等领域，作为微软“三屏一云”计划的核心技术，伴随着近期Windows Phone 7的发布，我们已能隐约感受到未来Silverlight“三端一统”的局面。

Silverlight发展至今，短短3年多仅为HTML5设计时间的一半就推出了四个主要版本，仍在研发中的第五个版本即Silverlight 5 的新特性已经公开，预计2011年春天会发布Beta版。Silverlight在其发展过程中充分吸收其他竞争对手的优点，取长补短，利用GPU硬件加速提升2D/3D应用程序性能，同时突破了Flash无法使用右键的功能限制，还能通过脱离浏览器的模式运行于系统桌面，以及作为Windows Phone等移动设备的主要开发技术，外加其开源框架Moonlight的跨平台特性，使得Silverlight在RIA领域的优势日益明显。不久，Silverlight 5或将带着对3D模型及骨骼动画的原生态支持问世，网页技术即将翻开崭新的一页——Web 3D时代的来临。

我们可以打这么一个比方，如果把Flash看做一只大盘股，那么HTML5就是只龙头股，厚积薄发的Silverlight则是名副其实的潜力股。就目前而言，这三者都在不断壮大中，各有各的优势与契机且相互制衡着，短时间内并不存在谁取代谁的问题。微软开发平台产品经理Brad Becker也曾说过：“在网络上，Silverlight从没打算要取代HTML5，而是要做HTML5和其它技术不能让开发人员轻易做到的事情。”

论发展速度，Silverlight有赶超Flash之势，HTML5则更多被用来满足某些厂商试图改变软件生态格局的发展战略需求。然而，所有的RIA技术都只为同一目的而存在着：让互联网绚动起来！RIA盛世降临，你准备好了吗？

作者简介 | 覃旋



中游在线技术顾问，微软最有价值专家(MVP)，微软社区精英，资深.NET技术讲师，Siverlight游戏引擎设计师。

■ 责任编辑：高松（gaosong@csdn.net）

三网融合下的国外互动式电视新契机

■ 文 / 柯博文

最近在国外掀起了一波精彩的机顶盒大战，笔者特地跟大家分享一下目前的情况与各个商品的优劣。

Apple TV

首先Apple 公司的Apple TV 值得一提。第一代的Apple TV的销售情况，连Steven Jobs 自己都不满意：定价过高、内容的来源全部来自iTune ，虽然后来还加入了YouTube，但是内容上还是少了一些。2010年9月，Apple再次推出了新版本的Apple TV——这款新的机顶盒，到底有何不同？目前来说，它的超低价格（99美元）、对iPad、Broadcast（透过WiFi）的支持已经引起大家的注意，但是市场的接受程度如何，还需要时间来检验。据使用者的反馈，现在已经发现一些有趣的东西：

第一，Apple TV 的OS 采用的是ISO 4.0 ，即它和Apple iPad 、Apple iPhone 采用了同样的OS。

第二，它居然携带有CPU！而且跟Apple iPad 是同一个款式。这就很有想像空间了。尽管一些骇客，早在第一时间内就把它给破解了，但透过目前圣诞节期间的销售情况，AppleTV已经是今年送礼最佳的选择之一。

说实在的，其实Apple TV 的功能并不多，也许这正是苹果公司的意图，可以使得它只专注在影像、音乐的部分——可以让用户通过iPad、iPhone十分方便地把音乐、影片及时地送到Apple



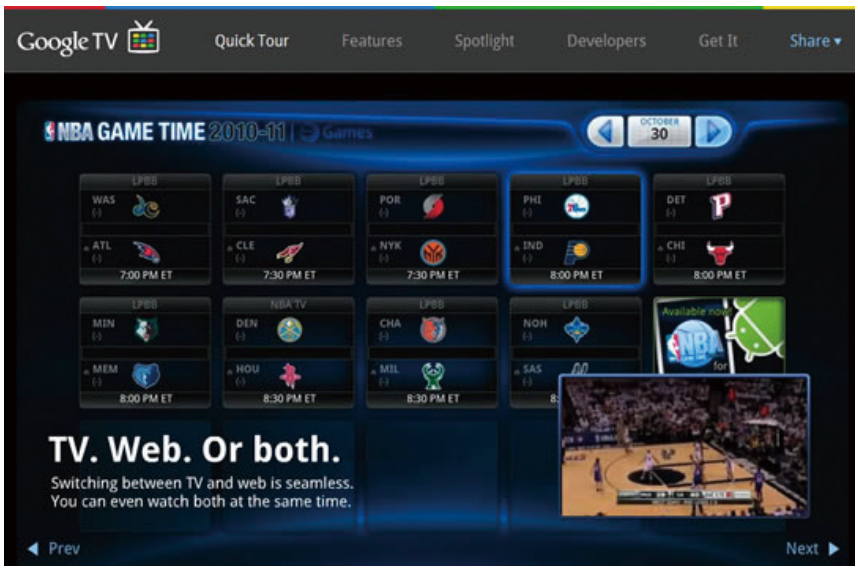
TV 上面，让用户直接可以通过大电视机屏幕观看。所以，这款产品并没有设置如互联网、应用程序软件、天气新闻等方面的功能。但正因为它专注于满足“通过电视看影片”的用户需求，而且画面高清、播放流畅，操作简单，加上价格便宜，所以目前至少在我认识的美国朋友们中，颇受欢迎。

TIVO

来自美国硅谷的TIVO，也是本次客厅大战的主角。这是一个以Linux为基础的Set-Top Box，运行软件均由Tivo公司自己研发，并且它也是一台首先提供EPG（电视节目表）和DVR使得电视节目可以暂停、倒转快转的机器。此外，你还可以把电视节目用硬盘存下来，当然这需要它是数字信息。TIVO起初在美国贩卖的时候，采用超低价格策略，外加mail- in rebate （即填写资料寄回公司，可以收到现金回馈），所以一经推出就迅速打开了市场。

Boxee

TV Boxee Box已经在2010年11月上市，它采用了Atom CE4100，是首款可播放网络电视、影片、音乐及照片的多媒体播放装置，支持目前最流行的数据格式与1080p高画质及5.1声道。现在友讯Boxee的内容提供商及应用服务有BBC, Last.fm, CNET, CNN,Channel 4, Comedy Central, Facebook, ITV, MTV



Music, mySpaceTV,YouTube, Flickr,and Picasa。

英特尔今年在IDF强打智慧型电视牌，力推Google TV，非Google TV联盟成员的美社群电视Boxee则推出了Boxee BOX力抗。友讯横跨G Tv与Boxee这两大联盟，无疑成为了智慧电视大战的大赢家。

Google TV

再看看Google。它最近致力的主要专案是“Google TV”。由于上回的手机Google Nexus One用的是单打独斗的行销方式，而且市场反应平平，所以这回Google TV 改变策略，联合多家公司打组合拳：包括使用Intel CPU Atom CE4100和Sony的HDTV与蓝光技术Blue-ray 播放器，并且使用Logitech的万用遥控器来控制画面。随着时间慢慢临近，现在Google TV的样子也越来越清晰了。根据目前的消息，它采用的机器是Logitech 的Revue，有一个HDMI的输入端；在输出端，它可以将Google TV软件，像个浮水印一样，叠在电视节目上面。





此外，Google TV 把眼光锁定在客厅电视机，并且把普通电视机通常拥有的几百台电视节目，一下子扩充到上万，这是如何做到的？当然用Google的强项——互联网，把网络上成千上万类似YouTube的网站连接进来，再加上开放式系统，把Hulu, Nextflix, Pandora 整合进来——这样还不够吗？它可以把你家里的DVR或个人媒体服务器（如电脑、智能型手机）的视频与音乐都连接到你家的电视机上。所以它就会象是家里的多媒体播放中心一样，包括一般的电视节目功能也都会具备。如果往底层看，这是一个Intel Atom CE4100（all-in one systejm-on-chip）的机器。是机器就一定需要有OS，猜猜会是哪一个？对！Google Android外加Embedded Linux。所以这是令人振奋的消息，意味着Android 智能型手机上目前的5万套软件通通都可以在Google TV上运行。哇！你可以想像在Android上受欢迎的手机软件全部都可以通过大屏幕的电视机来展现！但是，该如何操作使用它呢？这时Logitech 的强项就体现出来了，“Harmny”这一个结合万用遥控器加上一个touchpad 就可以做到类似像触屏那样的功能，来与Android Apps作互动了。如果看到这里，你只是把它当成Android 智能手机放大版的话，那就错了。还记得Google 的强项（互联网）吗？那里面的Browser，或者用更专业的技术来说，放在显示电视前面的选单，（如同现在一般电视的设定菜单），用的是“Chrome Browser”。这样你可以一边看电视，一边上网，或者是把画面切到网络上的节目。Chrome Browser可以让你去连接任何互联网网页。所以当Google TV 把电视和网页做成一个彼此沟通的平台后，一定会激起一个新的风暴——当电视遇到互联网，这里可想像的空间非常的大。如果你现在就想先试着写一些软件在Google TV上运行的话，HTML也许是目前最简单的方式。Google TV当然会提供API 以方便程序员可以设计专用的Google TV软件。但好消息是，Google TV 也会公开源代码，而且是免费的！

作者简介 | 柯博文



美国硅谷LoopTek公司CEO，专注于Google Android和Apple iPhone手机应用软件开发。

■ 责任编辑：高松（gaosong@csdn.net）

新产品&新工具

人大金仓推出BI统一平台

人大金仓“KingbaseBI统一平台”区别于人大金仓单一的数据库产品，它可为用户提供数据存储、数据抽取及过滤、数据仓库、数据挖掘及展示的整套解决方案，通过相关手段将业务数据转换成企业领导决策时的参考数据。“KingbaseBI统一平台”工作过程中，需要将大量业务数据通过KingbaseDI（ETL工具）进行数据抽取、数据清洗转换、数据加载，消除冗余后整合放在数据仓库中，最后通过在线分析的工具对数据进行分析和处理，以柱状图等丰富多样的报表工具展现出来。



云系统 Jolicloud 1.1 发布

Jolicloud OS 是一个基于Ubuntu专为上网本定制的操作系统，与Chrome OS类似，Jolicloud OS也定位于云系统，内置了大量的在线应用，它可以像Ubuntu一样直接通过Ubuntu官方源来安装本地软件。根据Jolicloud官方Blog的通告，最新的1.1版已正式发布，主要更新有新增自定义控制台和背景图片、简化对计算机的访问设置、能够从Jolicloud设备添加本地软件等。

JavaScript 文档利器：JsDoc Toolkit

JsDoc Toolkit是发布在Google Code上的一个开源项目，和其他语言的文档工具一样，它可以自动从 JavaScript 代码中提取注释生成格式化文档。运行环境 JsDoc Toolkit是用Java开发的，运行时需要 Java 1.5+。在运行之前，你需要把当前的工作目录切换到JsDoc Toolkit目录，并确保将java.exe所在目录添加到环境变量中。

下载地址：http://code.google.com/p/jsdoc-toolkit/downloads/list

RT-Thread 0.4.0 beta1发布

开源实时操作系统RT-Thread发布了0.4.x系列的第一个版本：自RT-Thread 0.3.0正式版发布以来，经过开发人员整8个月的密集开发，众多用户的积极反馈、测试，终于完成了RT-Thread 0.4.x系列的第一个测试版本。这个测试版本，依然更多地面向测试目的，不建议直接应用到实际的产品中。RT-Thread 0.4.0的非正式版本依然沿用GPL v2许可证进行发布。RT-Thread 0.4.0版本初始提出的几个功能亦基本达到，提供了相对完善的POSIX thread接口，动态库载入接口libdl以及应用模块支持。这些新功能的加入使得RT-Thread能够初步地兼容于POSIX标准。

Firesheep反制工具BlackSheep发布

Firefox扩展程序Firesheep，能监听和劫持WiFi热点或无保护网络中的未加密社交网站账号，开发者的本意是提醒Facebook、Twitter等网站采取措施保护用户。现在，Zscaler研究人员推出了反制工具BlackSheep。它也是一个Firefox扩展程序，会监视网络流量，如果发现网络中有Firesheep的活动痕迹，会警告用户。BlackSheep的工作过程是向网络中释放虚假的ID会话信息（Cookies），然后观察它是否被劫持，它甚至能提供入侵者的IP数据。

GParted 0.7.1 发布

GParted 是一款以直观的图形环境为硬盘划分分区的工具，与Partition Magic同样出色。GParted 0.7.1 提高对主板BIOS RAID的支持、修复dmraid分区路径名称错误、修复MBR中没有反映GPT的分区表，需要libparted-2.3来实现、更新翻译、修复Bug。

傲游手机浏览器正式版全球发布

傲游手机浏览器（Maxthon Moblie）正式版是国内首款通过云端技术实现跨平台同步功能的智能手机浏览器，用户可以随心所欲地在手机、平板、笔记本之间进行包括收藏在内的个人数据在线同步。

傲游手机浏览器正式版包含六大特色功能：同步收藏——通过傲游账户，可以同步电脑及手机上傲游浏览器中收藏夹内容；手势功能——通过画出简单手势轨迹即可完成实际功能操作；多标签浏览——多个页面一行标签，切换、关闭页面更方便；快速访问——自定义九宫格，收藏最常访问网址，上网更方便快捷；下载功能——支持断点续传、多格式文件下载；RSS阅读器——桌面小插件与浏览器结合，完成相关资讯订阅功能。

SafeNet推出新一代安全平台

2010年12月14日，SafeNet公司宣布其产品Luna SA 5.0硬件安全模块（HSM）在下一代PKI、云环境和虚拟环境中的可用性。通过利用SafeNet的技术专利即防篡改的ASIC数据安全保护处理器，敏感的关键材料能够很好地嵌入到硬件中，从而使这款新一代硬件安全模块（HSM）能提供更强的性能和防止地面撞击的安全性能。

Luna SA 5.0硬件安全模块具备基于硬件的密钥管理和分割的平台功能，提供了作为同一个虚拟化和云基础设施便利的内部部署的安全和控制的水平。它将被广泛用于保护公钥基础设施如供电电子发票使用、先进计量基础设施、安全生产等下一代PKI应用。

CI（持续集成）服务器Hudson

Hudson用起来很傻瓜，都是图形化界面，不需要自己去记忆各种命令然后写XML配置文件。Cruisecontrol明显太麻烦了，每次用都要找参考书找各种命令。看看Windows和Linux，就知道傻瓜使用方式才是王道。

RedHat企业版 4.9 Beta1 发布

RedHat Enterprise Linux 4.9 Beta1特性包括：使用2.6.9-92.EL内核、Red Hat Network （RHN）Channels测试网络、更新驱动、Firefox 3.6.4、OpenOffice 2.0办公系统、跟踪探测工具SystemTap更新到1.3，并对kprobe新功能支持、修复Bug等。

SAP Business ByDesign解决方案2.5版功能包

SAP Business ByDesign定位于成长型企业，为它们提供完善的管理支持。实现了实时、内存分析、支持移动终端、丰富的定制化用户界面，以及增强的灵活性等五大创新。

具体来说，SAP Business ByDesign 2.5提供成熟的客户生命周期管理功能，支持海量数据并降低了交付成本。除了全面支持多租户架构之外，该功能包还设计了专门面向客户和合作伙伴的业务扩展功能，同时允许用户灵活方便地修改默认的用户界面、报告及表单格式。基于SAP针对IT系统全生命周期管理的体系，该功能包将在升级过程中保护所有已实现的改动。

微型设计专用工具Dorado

腾讯CDC在官方博客上推出免费的微型设计专用Dorado。Dorado能够为设计稿进行尺寸标注及界面取色标注，适用于所有设计师、界面与网页前端设计开发者。

其特点包括：设计稿尺寸标注与取色专用工具，适用于设计、界面开发与网页前端；安装包仅700KB，全绿色；独有的双模式切换可支持双屏显示器，一面设计，一面标注；在全新S3D图像引擎技术下，支持PSD、PNG 、BMP、GIF、JPG、JPEG、JPE、JFIF、TIF、TIFF、EMF等17种图片格式文件；支持图像的复制、粘贴、导出，工具全键盘操作更是不在话下。







iRobot Scooba 380

iRobot Scooba是一个获奖的产品，目的是替代传统拖把。它可以自动帮你将房间地板清洗得干干净净。内部清洗系统包括吸尘、打清洗剂、刷洗和清理四个步骤，效果比拖把要好很多，还可以通过设置两个“虚拟墙”来限定清洗范围。一次充电可以使用两小时左右。

GEEK产品







iNo Mobile CP09

iNo CP09是一款专为老人或小孩设计的手机，特点非常鲜明：按键大、字体大、铃声大、待机时间长、操作非常简单。此外还带FM收音机和LED灯，而这些功能都是放置在手机两侧的功能键上，一推就可以直接使用。非常适合年龄大、视力不好以及对电子产品复杂功能望而却步的老人使用。

PogoPlug

PogoPlug是一款基于Linux和ARM的小型服务器，可以连接到互联网让用户共享USB驱动器的文件，甚至可以配置成一个小型LAMP服务器。应用时只需要简单地插上电源、接入网络，PogoPlug就会自动在其云端服务器进行认证登录，然后在任何有网络的地点访问你在PogoPlug上的数据。同时它还整合了Facebook、Twitter等服务。对于个人用户来说，PogoPlug 比一个NAS设备要划算得多。





抱枕遥控器

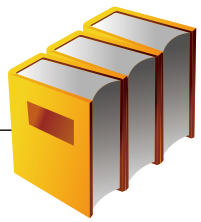
它是一个抱枕，也是一个遥控器。这个抱枕遥控器在抱枕中内置一个通用型号的可编程遥控器以及红外LED，可以用来遥控包括电视、DVD等500多种设备。抱枕表面上的按键标识下面都对应有一个软键，整个遥控器采用两节电池供电。



NES Electric Guitar

这把看起来酷酷的吉他实际上是由国外发烧友用老式任天堂主机改造而成的，采用标准的六弦吉他弦把，带有音量调节和节拍按钮。每把吉他都是全手工定制的。

新书上架



Effective C++：改善程序与设计的55个具体做法（第3版）



作者：Scott Meyers
译者：侯捷
出版社：电子工业出版社

书中的内容都是基础性的C++内容，其中介绍了55个准则，每一条准则描述一个编写出更好C++程序的方式。

本书相比前两版增加了两个新章，一是资源管理，一是模板编程。书中新素材还包括在异常概念下编程，套用设计模式，以及运用新的TR1程序库。

书中告诉大家在单线程系统中运作良好但可能不适用于多线程系统（Multithreaded Systems）的某些技术和做法。读者可以在附录B找到第二、三两版的条款对应表。本中译版与英文版页页对译，保留索引，偶尔加上小量译注。

MySQL技术内幕



作者：姜承尧
出版社：机械工业出版社

这是国内目前唯一一本关于InnoDB的著作。从源代码的角度深度解析了InnoDB的体系结构、实现原理、工作机制，并给出了大量最佳实践，能帮助你系统而深入地掌握InnoDB，更重要的是，本书完整地讲解了InnoDB存储引擎中最重要的一些内容，即InnoDB的体系结构和工作原理，并结合InnoDB的源代码讲解了它的内部实现机制。

全书首先全景式地介绍了MySQL独有的插件式存储引擎，分析了MySQL的各种存储引擎的优势和应用环境；接着以InnoDB的内部实现为切入点，逐一详细讲解了InnoDB存储引擎内部的各个功能模块；最后深入解析了InnoDB存储引擎的源代码结构，对读者阅读和理解InnoDB的源代码有重要的指导意义。

HMTL 5用户指南



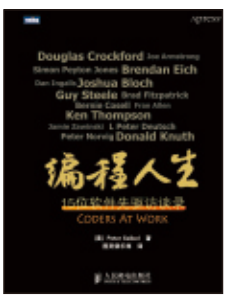
作者：Bruce Lawson、Remy Sharp
译者：刘红伟
出版社：机械工业出版社

作者并不以顺序的方式依次介绍每个元素或API，详细讨论每一个元素，之后再继续进行下一个。这些应该是规范的工作，但本书并非只是给出最基本的细节，而是教授如何使用每个元素或API，或者讲述它们如何在上下文中一起工作。随着新主题的讨论，作者将构建一些示例，并且提示读者当有需要注意的新内容出现时再返回来学习。

全书共分为10章，系统全面地介绍了HTML 5规范的核心内容，以及这些内容在当前浏览器中的支持情况，并告知开发者如何在当前的环境下应用这些功能，开发出漂亮的Web应用。

书中各章节短小精悍，但是信息量巨大；内容注重实践，其中的示例代码都具有很强的启发性和可操作性。

编程人生：15位软件先驱访谈录



作者：Peter Seibel
译者：图灵俱乐部
出版社：人民邮电出版社

这是一本访谈笔录，记录了当今最具个人魅力的15 位软件先驱的编程生涯。包括DonaldKnuth、Jamie Zawinski、Joshua Bloch、Ken Thompson 等在内的业界传奇人物，为我们讲述了他们是怎么学习编程的，在编程过程中发现了什么以及他们对未来的看法，他们对计算机软件行业的过去、现在和未来独到的看法和见解。他们对培养、发现、选拔、面试优秀的程序员的建议，并对诸如应该如何设计软件等长久以来一直困扰很多程序员的问题谈了自己的观点。

这本书并不是技术人士的八卦图书。书里讨论得更多是编程的习惯，对当今程序设计方法或是程序设计语言的讨论，以及这些编程大牛在成为大牛过程中的各种经历。很多东西都值得程序员或者是有志学习计算机科学的人借鉴。

结网@改变世界的互联网产品经理



作者：王坚

出版社：人民邮电出版社

做事情，不是仅仅知道如何去做就可以了，还需要知道为什么去做。作者结合案例介绍了产品经理如何去做，谈到了产品经理的职业发展和职业心态，更重要的是，透彻地分析了为什么要这么做。

全书以如何创建、发布、推广互联网产品为主线，介绍了互联网产品经理的工作内容以及应对每一部分工作所需的方法和工具。为用户创造价值是产品经理的第一要务，产品经理的工作是围绕用户及具体任务展开的，本书丰富的案例和透彻的分析道出了从发现用户到最终满足用户这一过程背后的玄机。本书面向现在正在从事及未来将要从事互联网相关工作的创业者和产品经理，也可以作为互联网产品策划人员或相关专业学生的参考书。新版完善了各章节，增加了优雅降级等内容，读者也可从中更深地去感受一名产品经理的感悟。

作者先后担任过QQ邮箱产品经理、Q吧产品总监、高级战略经理等职务，具有9年的互联网产品相关经验。于2005年创建个人网站糗事百科（qiushibaike.com），一个由用户分享真实糗事的社区，目前有90万以上的Feed订阅者。

iPad开发基础教程



作者：Dave Wooldridge Jack Nutting David Mark

译者：盛海燕

出版社：人民邮电出版社

2010年1月28日，iPad横空出世，毫无疑问，iPad是体验网络、收发电子邮件、翻阅照片、观看影片的绝佳选择。iPad重新定义了上网本和电子阅读器。如果正在寻找一本教你从头构建完整iPad应用程序的书，本书就适合。本书由知名移动开发专家执笔，带你开启激动人心的 iPad应用开发之旅。书中通过构建一个图形程序Dudel，介绍了iPad SDK的大量新特性，比如贝塞尔路径、PDF文件的创建、新框架Core Text、浮动窗口、视频播放控制器、拆分视图控制器、新的输入方法、手势识别和文档支持等。本书适合想构建iPad应用程序的iPhone开发人员阅读。

本书合著者Jack Nutting是资深Apple技术专家，Objective-C和Cocoa框架的坚定拥护者。除本书外，还与人合著了《Learn Cocoa on the Mac》。他有时会在www.nuthole.com上发表博客。

Dave Wooldridge是Electric Butterfly创始人，具有多年Mac或iOS应用程序开发经验，并且屡获殊荣。除本书外，还著有《The Business of iPhone App Development》一书。另外，他还为一些主流技术出版物撰写了大量文章，包括MacTech上每月一期的软件营销专栏。

David Mark是深受爱戴的Apple技术专家，具有多年开发经验。他著有畅销技术书《iPhone 3开发基础教程》和许多Mac平台编程参考书。可以通过www.davemark.com与他联系。

	Amazon
1	The LEGO MINDSTORMS NXT 2.0 Discovery Book: A Beginner's Guide to Building and Programming Robots
2	Getting Started with Arduino
3	The Facebook Effect: The Inside Story of the Company That Is Connecting the World
4	Head First Java, 2nd Edition
5	Presentation Zen: Simple Ideas on Presentation Design and Delivery
6	Hello, Android: Introducing Google's Mobile Development Platform (Pragmatic Programmers)
7	Switching to the Mac: The Missing Manual, Snow Leopard Edition
8	iPad For Seniors For Dummies
9	Droid X For Dummies
10	The Android Developer's Cookbook: Building Applications with the Android SDK

	天珑书局（中国台湾）
1	Google Android 2.X应用程序开发实战
2	深入浅出Android系统原理及开发要点
3	iPhone 创意程序设计家，2/e（适用 SDK 3、SDK 4）
4	Google Android SDK开发范例大全 2
5	ASP.NET 4.0专题实务－使用 C#
6	软件构筑美学：当项目团队遇上失控程序，最真实的解决方案
7	HTML 5 & API网页程序设计
8	全球最强VMware vSphere 4企业环境建构
9	App程序设计入门—iPhone&iPad
10	最严选！C#案例模块开发讲座

	第二书店
1	深入理解计算机系统（原书第2版）
2	领域驱动设计：软件核心复杂性应对之道
3	SEO实战密码——60天网站流量提高20倍
4	编程人生：15位软件先驱访谈录
5	互联网运营智慧——高可用可扩展网站技术实战
6	Android应用开发揭秘
7	微软应用架构指南
8	HTML5解密
9	程序员的思维修炼：开发认知潜能的九堂课
10	重构：改善既有代码的设计

用户的需求从哪里来？



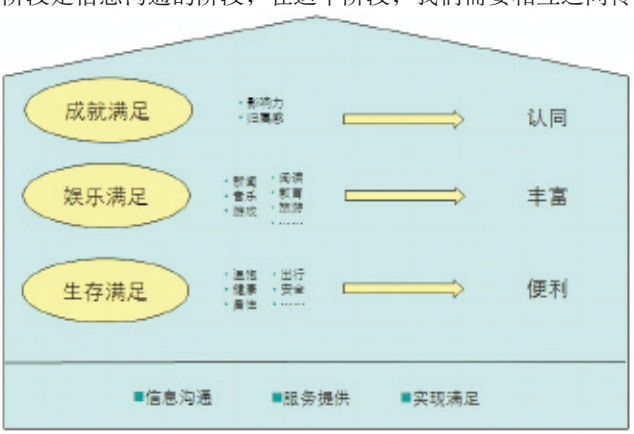
王煜全，Frost & Sullivan中国区总裁

记得曾经看过一个笑话：一个企业老总把他的IT管理人员叫到办公室说，“我在公司的电脑里按Ctrl+C之后，回到家里电脑按Ctrl+V，需要的东西怎么没有拷贝下来呢？不是说用Ctrl+C复制下内容来，然后用Ctrl+V就可以粘贴吗？”IT管理人员一脸无奈地告诉老总，在这种情况下，无论怎么Ctrl+C和Ctrl+V都不行！

我们也只是把它当成一个笑话来听，一笑而过。可是，有心的人就会考虑：为什么不行呢？这是用户的需求啊！

其实，人类自诞生以来，一直在向满足自己需求的方向迈进，这些需求满足可以简单分为三大类：生存满足、娱乐满足、成就满足。

为了完成这些需求的满足，必须经过三个阶段。第一个阶段是信息沟通的阶段，在这个阶段，我们需要相互之间传



递一些信息，这些信息是达成我们需求满足的必要基础，通过信息的沟通，我们知道如何去解决基本的温饱，如何去获得娱乐的信息，如何找到我们所属的圈子。第二个阶段是服务提供的阶段，在这个阶段，我们需要一些工具能够实现我们的信息沟通和传递，例如可以让我们快速进行信息交换的电话可以让我们找到丰富旅游信息的报纸。第三个阶段就是需求实现的阶段，在这个阶段，人们通过所提供的服务，达成最终的需求满足。

所以，用户的需求就是从我们的身边生活中来，可能就是我们身边一个个不起眼的要求，存在于我们可能会忽视的地方，例如文章开始那位企业老总的需求！

需求的出现和完成也是随着时代的进步而一步步前进的。当我们需要快捷地和远方的人进行交流的时候，我们发明了电话；当我们需要随时随地找到自己想要找的人的时候，我们发明了移动通信；当我们需要快速运算的时候，我们发明了计算机；当我们需要把计算机的力量汇集起来的时候，我们发明了互联网；当我们需要通过互联网来寻找相关信息的时候，我们发明了互联网网站；当我们需要从海量信息中查找合适内容的时候，我们发明了搜索工具。那么，当我们有人需要在办公室Ctrl+C，回家进行Ctrl+V的时候，我们需要发明什么呢？

这就是未来无线互联网发展中的一个巨大需求：用户相关内容的存储转发服务。

企业老总在公司里通过Ctrl+C命令，将自己需要的内容存储起来。存储在哪里呢？这就是我们服务提供的关键点，将内容通过无线网络存储在远端服务器，用当下时髦的话来说，就是存储在“云端”。等回到家后，通过Ctrl+V命令，将存储在云端的内容下载到任意的设备中。

用户需求有了，我们该怎么去实现呢？这就需要我们技术人员构建系统的服务能力，而且必须是用户使用方便的服务能力。我们可以知道，目前是有一些类似的存储服务能力，但是操作起来还是比较复杂，用户都希望傻瓜似的操作，就像老总就会Ctrl+C和Ctrl+V！

而且，这个需求必定会延伸，假如有一天，企业老总说：为什么我在这里Ctrl+C后，我家人在家里Ctrl+V就不行呢？我们也有办法。

一则笑话给我们带来的却是一个用户需求的深化。有可能给我们带来一个前途光明的服务。这个服务可是具有很大的价值。我们目前几乎每个用户都用运营商提供的短信服务，每条短信要花1毛钱，不管你是写1个字，还是写70个字。如果我们有这样的服务，而且操作和短信一样简单，那么好了，我们不妨打打价格战，于是造福了广大群众，同时也可能造就新的企业。因为，我们不只是简单发发短信，而是能够转发一切用户需要的内容。



高巍，安卓爱普公司创始人。原搜狐媒体产品中心经理。关注移动互联网、互联网产品管理。
微博：
httpt.sina.com.cn/inetpm

程序员是这个时代的思想家

好吧，这句话其实是3G门户创始人张向东说的，“这个时代的思想家是程序员。”当然，你可以争辩，话反过来说就是另一层意思。标题是有点哗众取宠，那就改成“程序员更有机会成为这个时代的思想家。”

张向东在评论KK（凯文·凯利，《连线》杂志创始主编）的著作《失控》时，提出了这个观点。KK作为未来学家，经常强调的却是，要了解未来必先了解历史，从历史中能看到未来的趋势。《失控》以及KK新书《技术想要什么》都是在一个宏大的数万年甚至百万年的生命进化史的叙事背景下展开的。“时代的思想家”这个命题，也不妨拉升到历史的广角视野中考察。

KK的一个观点是，人类与人类的创造物——技术，是共同进化，相互依存。当人类驯化了狗、奶牛、谷物，人类自身也被驯化了。我们的牙齿不再锋利，我们的肌肉不再健壮，我们的毛发不再浓密。我们在创造技术、创造工具的同时，我们自身也在被再造。我们被自己创造的技术、工具所“驯化”和“再造”。

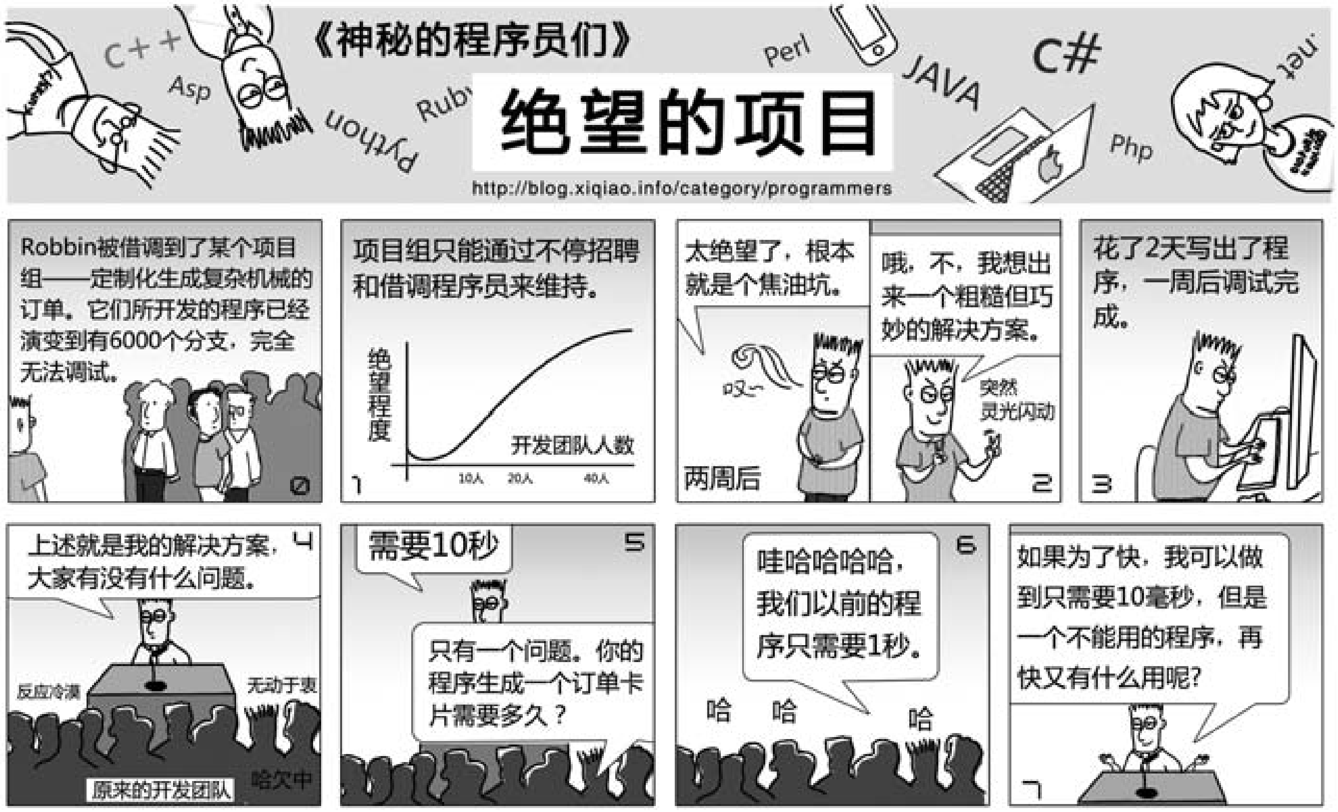
古代，思想家常常是文学家。小学课本中对孔子、庄子等大师的介绍，经常是“伟大的思想家、文学家……”充满哲理思辨的篇章，大多在文学上也很出彩，原因恐怕不仅仅是“言而无文，行之不远”那么简单。《技术想要什么》第一部分也许给出了一些线索。语言、文字的发明，是远古时代人类最伟大的技术创造，语言、文字是古代人类更为重要、复杂的技术工具。按KK的“人与技术共同进化”观点，技术使我们的思想、身体和基因发生改变。语言文字的娴熟运用、思想的进化，就会出现一个有趣的共生现象。

近代，思想家常常是数学家、物理学家。数学家拉普拉斯，更为有名的是机械决定论的世界观——宇宙就像一个精确运转的钟表的比喻。牛顿的经典物理著作，取名《自然哲学的数学原理》。哲学系的课程，一般包括近代物理。上了年纪的物理学家，发挥余热做科普演讲的主题，往往是“物理与哲学”。KK提到，技术在我们的生命中究竟意味着什么？面对永不停歇、日新月异的技术变革，我们如何找到“自我”的“锚点”？要回答这些问题，技术进化给了我们一个思考的框架。对数学和物理的谙熟，无疑有益于这种思想上的探索和参照。

这个时代，是Internet狂飙突进的年代，是KK所说的由“原子隐喻”到“网络隐喻”的年代。20世纪80年代，“四个现代化”的标志是卢瑟福的原子模型。今天，科幻电影关于未来世界的图景描述是Matrix。KK预言了“The One”，全世界20多亿台上网的个人电脑，加上超过40亿部手机，再加上所有的数据服务器，这些数十亿的机器通过有线和无线的网络连接起来，耗去了这个星球5%的电力，构成了一台庞大的虚拟机器。有些人称之为“云”，有些人称之为“外脑”。无论如何，人类将要与这样一个“超级网络”共生共存、共同进化。这个“超级网络”将给予人类更多的选择、可能性，深刻影响人类社会的进化。

那么，谁更有机会成为这个时代的思想家？







Robbin: Unix程序员。AppleFans。Geek。急需女朋友。命令行狂热者。



Hank: Windows程序员.NET高手。已婚。保守派。



Ada: 架构师。第三性别，喜欢挖苦男人。热爱无所不能的emacs。

作者介绍：西乔
项目经理。06年起携创业团队从事Web技术外包开发及产品咨询顾问。



如果你有什么好玩的关于程序员的故事、对话、代码，愿意通过漫画的形式分享，请给西乔发邮件：arthur369@gmail.com。

幽　默

1. 某同学决定以GPL协议发布他的作业，值得注意的是，他是年级第一名。
2. 文学课上，老师问同学，你们看过《边城》吗？同学回答：我们不是计算机系的。
3. 一名IT经理走进拉面馆说：“请问，你们需要客户端吗？”老板说：“面一般是伙计端，忙的时候才需要客户端。”
4. 中国一留学生去美国打工当报童，不带计算器，习惯动作抬头望天时心算找零。顾客大为惊讶，纷纷掏出计算器验证，皆无误，也抬头望天，惊恐问：“云计算？”
5. 有两种生成一个软件设计方案的途径。一个是把它做得如此简单，以至明显不会有漏洞存在。另一个是把它做得如此复杂，以致不会有明显的漏洞存在。
6. 《爱情买卖》之设计师版：
   设计师对着产品经理唱的神曲：当初是你要修改，修改就修改；现在又要拍脑袋让我改回来；设计不是你想改，想改就能改；让我改版让我修改你丫自己来！
7. 程序员借钱
   X：哥们，最近手头紧，借点钱？
   Y：成啊！要多少？
   X：一千行不？
   Y：咱俩谁跟谁！给你凑个整儿，这一千零二十四，拿去吧。
8. 一直觉得NVIDIA应该把GeForce更名为GeiForce，中文翻译成“给力”。
9. 上微机课，用VC编程，X坐Y旁。
   X：为什么你的关键字会变蓝而我没有？
   Y看了一下，没来得及说话。
   X又问：为什么我有的字会有波浪线而你没有？
   Y又看了看X的电脑说：为什么你要用Word来编程？
10. 盖茨夫人接受采访时说：我们家从来不用苹果的产品，甚至连苹果都不吃。坐在一旁的乔布斯不屑一顾地说道：这有什么了不起的，我们家连窗户都没有。
11. 锄禾日当午，不如Coding苦，对着C++，一调一下午；
    锄禾日当午，不如Coding苦，调了一下午，Bug还得补；
    锄禾日当午，不如Coding苦，Bug刚补完，结构需重组；
    锄禾日当午，哪有Coding苦，Bug改不完，无颜见列祖。
12. 据说某俄国特工九死一生偷到了NASA太空火箭发射程序源代码的最后一页，代码是：} }